Software Detailed Design Document

Development of the Address Register Information System

EDMITE PROJECT

Chisinau, 2017

Table of Contents

List of tables

List of figures

1 Introduction

Based on the strategy on address system implementation and legislative framework, currently under endorsement by Parliament, this project aims the development of Address Register Information System (ARIS) for the Republic of Moldova, under the Agency of Land Relations and Cadaster (ALRC) and State Enterprise "Cadastru".

In this regard, Address Register Information System is a component part of the National Geographic Information System and represents a register that contains classifiers of administrative-territorial units of Republic of Moldova (districts, cities (municipalities), villages (communes), localities, including disbanded cases) and basic elements of urban infrastructure (streets, segments, buildings, entrances, isolated premises) qualified as addresses of physical objects.

ARIS contains the identifier for objects of evidence and their basic characteristics (name, formation date, liquidation date, etc.), as well as borders of administrative-territorial units, streets axial lines and buildings outlines, entrances position and exact address localization.

ARIS is intended for identifying elements of urban and rural infrastructure, considered physical objects, which are objects of evidence for departmental and interdepartmental information systems and that establish links with postal address system, used by information systems, as well as juridical and physical persons in daily life.

Along with information from National Geospatial Data Fund (http://www.geoportal.md) and from automated information system "Cadastrul bunurilor imobiliare", ARIS ensures spatial localization and identification of any kind of object of the address in the basic spatial Model of the terrain.

2 References

The processes concerning the creation, implementation of ARIS is not contravening the field-related regulatory acts in effect regarding the S.E. „Cadastru" activity and the development of IT solutions intended for the Moldovan public authorities.

This category comprises the following legal and regulatory acts:

1. Concept of the State Automated Information System „Address Register".
2. Government Decision No. 710 of 20.09.2011 on approving the Strategic Program for Technological Modernization of Governance (e-Transformation), "Monitorul Oficial al Republicii Moldova" No. 156-159 of 23.09.2011.
3. Government Decision No. 656 of 05.09.2012 on approving the Interoperability Framework Program, "Monitorul Oficial al Republicii Moldova" No. 186-189 of 07.09.2012.
4. Government Decision No. 1090 of 31.12.2013 on electronic governmental service of authentication and access control (MPass), "Monitorul Oficial al Republicii Moldova" No. 4-8 of 10.01.2014.
5. Government Decision No. 405 of 02.06.2014 on integrated electronic governmental service of digital signature (MSign), "Monitorul Oficial al Republicii Moldova" No. 147-151 of 06.06.2014.
6. Government Decision No. 708 of 28.08.2014 on electronic governmental service of logging (MLog), "Monitorul Oficial al Republicii Moldova" No. 261-267 of 05.09.2014.
7. Government Decision No. 916 of 06.08.2007 on the Concept of Governmental Portal, "Monitorul Oficial al Republicii Moldova" No. 127-130/952 of 17.08.2007.

8. Government Decision No. 330 of 28.05.2012 on creation and administration of single governmental portal for public services, "Monitorul Oficial al Republicii Moldova" No. 104-108 of 01.06.2012.
9. Law No. 133 of 08.07.2011 on Protection of Personal Data, Official Gazette No. 171-175 of 14.10.2011.
10. Government Decision No. 1123 of 14.12.2010 on approving the Requirements for the assurance of personal data security during their processing within the information systems of personal data, Official Gazette No. 254-256 of 24.12.2010.
11. Government Decision No. 701 of 25.08.2014 on approval of Methodology of publishing open governmental data, "Monitorul Oficial al Republicii Moldova" No. 256-260 of 29.08.2014.
12. Law No. 264-XV of 15.07.2004 on electronic document and digital signature, "Monitorul Oficial al Republicii Moldova" No. 132-137/710 of 06.08.2004.
13. Government Decision No. 945 of 05.09.2005 on Centers for Certification of Public Keys, "Monitorul Oficial al Republicii Moldova" No. 123-125 of 16.09.2005.
14. Government Decision No. 320 of 28.03.2006 on approving the Regulation on applying digital signatures in public authority electronic documents, "Monitorul Oficial al Republicii Moldova" No. 51-54 of 31.03.2006.
15. Government Decision No. 735 of 11.06.2002 on Special Telecommunications Systems of the Republic of Moldova, "Monitorul Oficial al Republicii Moldova" No. 79-81 of 20.06.2002.
16. Law No. 467-XV of 21.11.2003 on Informatization and state information resources, "Monitorul Oficial al Republicii Moldova" No. 6-12/44 of 01.01.2004.
17. Standard of the Republic of Moldova SMV ISO CEI 15288:2009, "Systems and Software Engineering. Processes of the system life cycle".
18. Technical Regulation "Processes of software life cycle" RT 38370656-002:2006; "Monitorul Oficial al Republicii Moldova" No. 95-97/335 of 23/06/2006.
19. Other laws, regulatory acts and standards in force in the ITC area.

The international guidelines and recommendations listed below have been implemented in order to define ARIS concept and ensure its further development:

1. INSPIRE Metadata Implementing Rules: Technical Guidelines based on EN ISO 19115 and EN ISO 19119, http://inspire.ec.europa.eu/documents/Metadata/MD_IR_and_ISO_20131029.pdf
2. INSPIRE Data Specifications on Address Data Model: D2.8.I.5 Data Specification on Addresses-Technical Guidelines, http://inspire.ec.europa.eu/documents/Data_Specifications/INSPIRE_DataSpecification_AD_v3.1.pdf
3. INSPIRE Download services, View services, Discovery Services, Transformation services, http://inspire.ec.europa.eu/index.cfm/pageid/5
4. Michael O. Leavitt, Ben Shneiderman, Research-Based Web Design & Usability Guidelines, U.S. Government Printing Office, http://www.usability.gov/guidelines/guidelines_book.pdf
5. Recommendations of the World Wide Web Consortium (W3C) (http://www.w3c.org) on the quality of websites, the possibilities to have proper information visualization, using widely used Internet WEB browsers, and compatibility with different IT platforms;
6. Recommendation of the W3C (http://validator.w3.org) on website testing. All pages generated by ARIS shall be tested as per these recommendations.

3 Acronyms and Abbreviations

The totality of Acronyms and Abbreviations used in this document:

No.	Abbreviation/Acronym	Description
1	ARIS	Address Register Information System
2	ALRC	Agency of Land Relations and Cadaster
3	BPMN	Business Process Model and Notation
4	CEC	Central Election Commission
5	CPA	Central Public Authority
6	CSW	Catalog Service for the Web
7	DB	Database
8	GIS	Geographical Informational System
9	IT	Information technology
10	ITC	Information technology and communications
11	ITS	IT system
12	KPI	Key performance indicators
13	LPA	Local Public Authority
14	NFC	National Fund of Cartography
15	OLAP	Online Analytical Processing
16	RDBMS	Relational Database Management System
17	SAISE	State Automated Information System" Elections"
18	SDD	Software design document
19	SDI	Spatial Data Infrastructure
20	S.E." Cadastru"	State Enterprise „Cadastru"
21	SRS	Software Requirements Specification.
22	SRV	State Register of Voters
23	TLS/SSL	TLS Protocol or its predecessor, SSL Protocol, are cryptographic protocols that ensure reliable communication between two hubs of the computer network.

No.	Abbreviation/Acronym	Description
24	WMS	Web Map Service
25	WFS	Web Feature Service

Table 1 Acronyms and Abbreviations

Terms frequently used in this document:

No.	Term	Description
1	Address	Totality of words, numbers, orthographic signs, placed in a certain order, indicating the exact geographic position of the addressable object
2	Credentials	A set of symbols that establish the users' and systems identity and authentication within information systems.
3	Data	Elementary information units about people, subjects, facts, events, phenomena, processes, objects, situations, etc. presented in a way that enables their notification, commenting and processing.
4	Database	A collection of data organized as per the design structure describing the basic characteristics and relation among entities.
5	Data integrity	Data status when they maintain their content and are interpreted unambiguously in cases of random actions. It is deemed that the data maintained their integrity if they have not been altered or deteriorated (deleted).
6	Information and Communications Technology	Common term that includes all technologies used for information exchange and processing.
7	Information object	Virtual representation of existing tangible and intangible entities.
8	Information system	A system for information processing along with the associated organizational resources such as human and technical resources, which deliver and disseminate the information.
9	Information resource	Set of documentary information in the IT system, maintained as per the requirements and legislation in force.
10	IT system	The totality of software and hardware that ensures data automatic processing (the automated component of the information system).
11	Logging	A function of recording the information on events. The records about events entered into the information systems include details about the date and time, user, and action carried out.

Table 2 Used Terms

4 Implementation method

In order to ensure the attainment of the objectives set for the IT solution, the following general principles have been taken into account while designing, developing and implementing ARIS:

- Principle of Legality: implies the establishment and operation of Information Systems in compliance with the national legislation in effect and with the international rules and standards recognized in this area;
- Principle of split-level architecture: involves independent design of ARIS components in compliance with interface standards between levels;
- Principle of service-oriented architecture (SOA): involves dividing the application operation into smaller and distinct units – called services – that can be assigned into a network and can be used together to create applications designed to the implementation of IT System business functions.
- Principle of reliable data: stipulates that data shall be entered into the system through authorized and authenticated channels only;
- Principle of information security: implies ensuring an adequate level of integrity, selectivity, accessibility and efficiency to protect the data against losses, alteration, deterioration and unauthorized access.
- Principle of transparency: implies designing and implementing as per the modular principle, having used transparent standards in the area of IT and telecommunications;
- Principle of expansibility: stipulates the possibility to expand and supplement the Information System with new functions or improve the existing ones;
- Principle of first person/single center priority: implies the appointment of a high-rank responsible person who has sufficient rights to take decisions and coordinate the activities aimed at Information System establishment and operation;
- Principle of scalability: implies ensuring constant IT performance when increased volume of data and stress for the Information System;
- Principle of usage simplicity and complacency: implies the design and implementation of all applications, hardware and software resources available to the System users, based exclusively on visual, ergonomic and logical principles of design;
- Principle of data integrity, fullness and reliability: implies the implementation of mechanisms that enable preserving the data content and their clear interpretation under cases of accidental influence, and elimination of data distortion or accidental liquidation, delivery of data volume sufficient to perform the IT System business functions and ensure advanced matching of data with the real status of objects they represent and belong to a specific sector of the IT System.

In particular, the following essential principles shall be complied with by the Information System Architecture:

- implementing a WEB based client-server solution with authorized access to interface and data;
- ensuring adequate security for the Information System to protect the information and subsystem components against their illegal use or disclosure of personal data or of information with limited access;
- recognizing information as an asset and ensuring its proper management;
- developing and implementing Information Systems that enable their use for other processes or ensure opportunities for developing new functionalities;

- minimizing the number of various technologies and products that offer the same or similar functionalities as per their purpose (reuse of technologies already implemented within the S.E. „Cadastru");
- ensuring high-speed processing of service requests/inquiries addressed to the S.E. „Cadastru" or of other
- LPA or CPAs requiring services;
- ensuring recovery capacities following disasters (ensuring physical and logical security) as a component of the implementation plan.

The main purpose of „ARIS" is to produce a performance platform for creation, storage, modification, visualization and providing address data. Solution will serve as a single address book for Moldova based on the principle of a unique storage and free distribution on necessity.

Given solution provides a state-level interoperable environment that corresponds to the principles and ideas promoted by e-Government Center.

The development of "ARIS" achieves the following aims:

- Implement a unique state-level evidence of data on addresses;
- the development of a unique mechanism to add, modify, delete data for all territorial administrative units;
- the development of a unique high-performance repertoire of address data provision;
- create an interoperability framework among the S.E. „Cadastru" IT applications, external IT Systems that provide and use ARIS data;
- implement an efficient collaboration mechanism among all actors involved in registration and management procedures on addresses data;
- reduce the required time and the laboriousness of the process of collecting, processing and managing the data addresses of the Republic of Moldova;
- ensure data access control and maximum security and confidentiality to data collections and users;
- provide informational support to filed-related analysis, forecast and research activities.

4.1 Introduction

There are four fundamental phases in most, if not all, software engineering methodologies. These phases are *analysis*, *design*, *implementation*, and *testing*. These phases address what is to be built, how it will be built, building it, and making it high quality. These phases will now be defined as they apply to the life cycle stage of product delivery emphasized in this thesis.

4.2 The Analysis Phase

The analysis phase defines the requirements of the system, independent of how these requirements will be accomplished. This phase defines the problem that the customer is trying to solve. The deliverable result at the end of this phase is a requirement document. Ideally, this document states in a clear and precise fashion what is to be built. This analysis represents the “what” phase. The requirement document tries to capture the requirements from the customer’s perspective by defining goals and interactions at a level removed from the implementation details.

The requirement document does not specify the architectural or implementation details, but specifies information at the higher level of description. The problem statement, the customer’s expectations, and the criteria for success are examples of high-level descriptions. There is a fuzzy line between high-level descriptions and low-level details.

The requirement descriptions of the things in the system and their actions does not imply an architecture design rather a description of the artifacts of the system and how they behave, from the customer’s perspective. Later, in the design phase, these requirement descriptions are mapped into computer science based primitives, such as lists, stacks, trees, graphs, algorithms, and data structures.

4.3 The Design Phase

In the design phase the architecture is established. This phase starts with the requirement document delivered by the requirement phase and maps the requirements into an architecture. The architecture defines the components, their interfaces and behaviors. The deliverable design document is the architecture. The design document describes a plan to implement the requirements. This phase represents the “how” phase. Details on computer programming languages and environments, machines, packages, application architecture, distributed architecture layering, memory size, platform, algorithms, data structures, global type definitions, interfaces, and many other engineering details are established. The design may include the usage of existing components.

In our approach, the team, given a complete requirement document, indicates critical priorities for the implementation team. A critical implementation priority leads to a task that has to be done right. If it fails, the product fails. If it succeeds, the product might succeed. At the very least, the confidence level of the team producing a successful product will increase. This will keep the implementation team focused. Exactly how this information is conveyed is a skill based on experience.

4.4 The Implementation Phase

In the implementation phase, the team builds the components from scratch or by composition. Given the architecture document from the design phase and the requirement document from the analysis phase, the team should build exactly what has been requested, though there is still room for innovation and flexibility. For example, a component may be narrowly designed for this particular system, or the component may be made more general to satisfy a reusability guideline. 2 The architecture document should give guidance. Sometimes, this guidance is found in the requirement document.

The implementation phase deals with issues of quality, performance, baselines, libraries, and debugging. The end deliverable is the product itself.

4.5 The Testing Phase

Simply stated, quality is very important. Many companies have not learned that quality is important and deliver more claimed functionality but at a lower quality level. It is much easier to explain to a customer why there is a missing feature than to explain to a customer why the product lacks quality. A customer satisfied with the quality of a product will remain loyal and wait for new functionality in the next version. Quality is a distinguishing attribute of a system indicating the degree of excellence.

In many software engineering methodologies, the testing phase is a separate phase which is performed by a customer team after the implementation is completed. There is merit in this approach; it is hard to see one's own mistakes, and a fresh eye can discover obvious errors much faster than the person who has read and re-read the material many times.

Internal testing deals with low-level implementation. Here each function or component is tested. This testing is accomplished by the implementation teams. This focus is also called clear-box testing, or sometimes white-box testing, because all details are visible to the test. Internal limits are tested here.

4.6 The Migration Phase

The objective of the Data migration process is to migrate, convert and test all existing data that is necessary for testing and for the operation of the new application. The Migration Strategy identifies sources and targets of the data migration and describes an overall approach to migrate, convert and test all existing data. The Migration Strategy shall address all important issues, e.g.:

- checking and testing data migration;
- migrating of historical data;
- inconsistency among data import;
- managing fractional, corrupt, etc. data records; converting of the classifiers in new system;
- indicating status of the migrated data to distinguish them from new records.
- Elaborate quantitative and semantic quality control procedures, define format of quantitative and semantic quality control reports. Quantitative and semantic quality control reports must be understandable for average user and clearly show the migration quality level

4.7 Methodology

The WaterSluice software engineering methodology separates the important aspects from the less important and concentrates on solving them first. As the process continues, finer and finer details are refined until the product is released. The WaterSluice borrows the iterative nature of a cyclical methodology along with the steady progression of a sequential methodology.

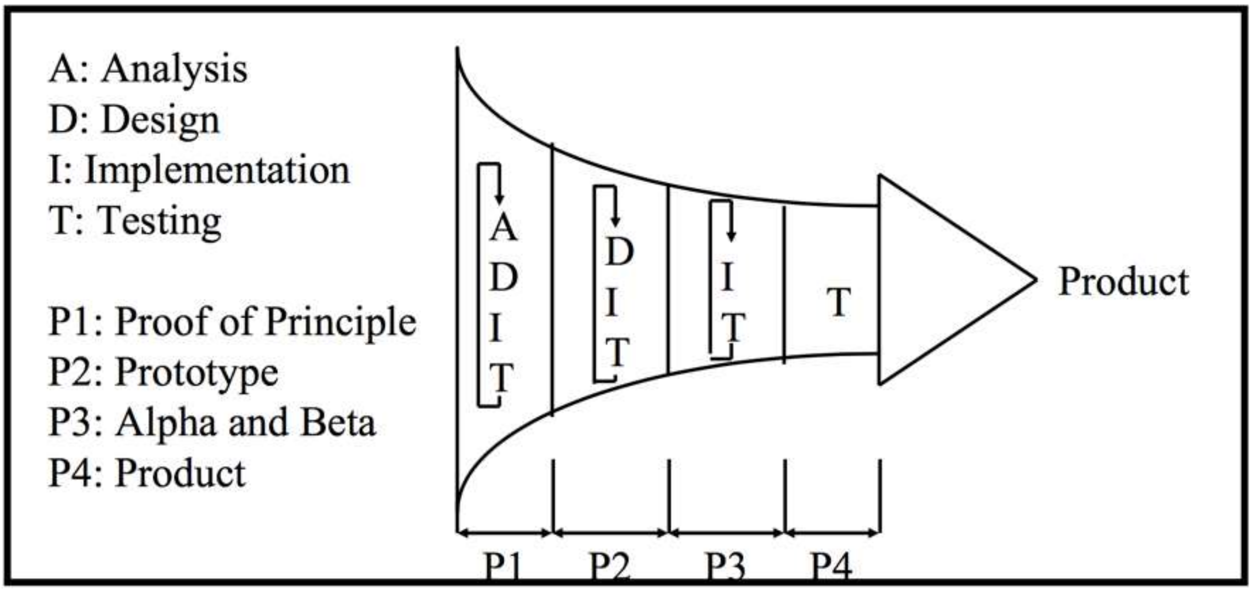

Figure 1 Methodology

At the beginning of the project, in an iterative process, the analysis, design, implementation, and test phases are broken into many potential tasks yet to be accomplished by team members. Each potential task is assigned a priority by team members. This priority reflects the benefit to the final goal of accomplishing the task based on what has already been accomplished. The highest priority task is accomplished next. De- pending on the size of the team, multiple high priority tasks may be accomplished in parallel. The remaining, lower priority tasks are held for later review. Exactly how many tasks or the granularity of the tasks is dependent on the size of the project, the size of the team building the project, and the scheduled delivery time for the project.

As a result of accomplishing these tasks, new analysis, design, implementation, or testing tasks may be discovered. These newly discovered tasks are then added to the known remaining open tasks and again prioritization is required. The next highest priority task are then accomplished.

Defining the priority function is of high importance. This priority function is domain-specific as well as institution-specific, representing trade-offs between quantity and quality, between functionality and resource constraints, and between expectations and the reality of delivery. The priority function orders the different metrics and their values. However, all priority functions should have the product delivery as a high priority goal.

The priority function serves two goals. One goal is to establish priority. Important tasks need to be accomplished first over lower priority tasks. This is the traditional role of a priority function.
The second goal of the priority function is to manage conflicting and non-monotonic tasks. The priority function needs to divide the tasks into consistent collections. The priority function needs to guide the selection of the consistent collection and then followed by the selection of the tasks within that consistent selection.

Once a component is completed to the satisfaction of the team, it is placed under change-order control. When a component is placed under the change-order control process, changes to the component are now frozen. If a change is absolutely necessary, and the teams are willing to delay the project to enforce the consequences of the change, then the change is fulfilled. Changes should be few, well justified, and documented.

Obviously, early in the process, analysis tasks are naturally a high priority. Later in the process, testing and quality become a higher priority. This is where the change-order control process becomes important. At the beginning of the process all four categories of analysis, design, implementation, and testing are available for prioritizing and scheduling. At the P1-P2 transition point, in the process, the analysis phase is subjected to change-order control process. Having the analysis phase frozen focuses attention on the remaining three categories of tasks. In a similar fashion, at the P2-P3 transition point, the design phase is frozen and at the P3-P4 transition point the implementation phase is frozen. At the final stage only changes that affect quality are allowed. This leads to a definition of temporal stages in the methodology, specifying priorities.

The main stages are called proof-of-principle, prototype, alpha and beta release, and product. With the exception of the proof-of-principle stage, these stages should not be new concepts to software engineers. The proof-of-principle stage represents the more traditional specification stage. Rapid prototyping offers a similar proof-of-principle stage.

In the first stage, the teams work simultaneously on all phases of the problem. The analysis team generates requirements. The design team discusses requirements and feeds back complexity issues to the requirement team and feeds critical implementation tasks to the implementation team. The testing team prepares and develops the testing environment based on the requirements.

The implementation team has to be focused on the critical tasks which is usually the hardest task. This contrasts the common practice of doing the simple things first and waiting until late in the product implementation to tackle the harder tasks. Most products that follow this practice end up failing. Once the critical task components have been implemented, the system, still a child in the first period of life, is ready for transition to the prototype stage.

In the second stage, the prototype stage, the requirements and the requirement document are frozen and placed under change-order control. Changes in requirements are still allowed but should be very rare. Any new requirements after this point are very costly. Only if the requirement change is absolutely necessary to the success of the product, despite the potential delays in the product delivery or cost over-runs, is the requirement change allowed. The main idea is to force control on any new requirements. This forces the cycle to be completed and enables product delivery. The architecture is still allowed to vary a little as technology pressures deliver new options.

Once the critical tasks are done well, the implementations associated with the critical tasks are expanded to cover more and more of the application.

One of the goals of this stage is for the team to convince non-team members that the solution can be accomplished.

At the end of this stage, the process is ready for transition into the alpha and beta release stages.

In the third stage, the architecture is frozen and placed under change-order control. This means that no more architectural changes are allowed unless they are absolutely necessary. Emphasis is now placed on the implementation and quality assurance.

The first version in field release is usually called an alpha release, while a second release is called the beta. The product may be immature in the alpha release. Only critical tasks have been implemented with high quality. Usually, only a limited number of customers are willing to accept an alpha version of the product and assume the associated risk.

During the beta release, enough of the system should be working to convince the customer that soon the beta application will be a real product. The beta release is more mature and is given to a much larger customer base.
When enough of the system is built, the system is ready for a transition into the next stage: releasing a high-quality product.

In the fourth stage, the implementation is frozen and focus is primarily on quality. At the end of the stage, the product is delivered.
One of the goals of the last stage is to make the product of high quality. No known critical errors are allowed in the final product. Sometimes, there is a gray area of definition between a product feature and a product error with the provider of the product, most often then not, providing features, while the customers viewing some features as errors.

The WaterSluice allows for phase interactions while at the same time setting firm temporal deadlines. The WaterSluice forces all four phases to communicate up front and to work together.

The WaterSluice software engineering methodology assumes the presence of five levels in a supporting software engineering environment as described. Versioning is used to move the product from one version to another version by repeating the methodology for each version. Risk management is assumed throughout the process. The major components of analysis, the details in the design phase, the four main phases of implementation, and levels of testing proceed as previously described.

4.8 Change-Order Control

Change-order control is a software engineering process that manages change, or lack thereof. The process is weighted to prevent change. Tools help to manage this process, while senior decision makers accept or decline change decisions. Frequently, the senior decision makers are independent of the teams.
Once a component is completed to the satisfaction of the team, it is placed under change-order control. When a component is placed under the change-order control process, changes to the component are now frozen. If a change is absolutely necessary, and the senior decision makers are willing to delay the project to enforce the consequences of the change, then the change is fulfilled. Changes should be few, well justified, and documented. Many change requests are postponed and incorporated into the next version of the product. Some of these change requests contribute to the requirement document for the next version, while some contribute to the architecture and implementation. Still, others may improve the quality.

5 Application architecture

ARIS has been implemented with the following layers of development and application frameworks:

Layer	Application	Plugins/Libraries	Requirements	Licensing
Development IDE	Eclipse Neon Camunda Modeler 1.11.2	-	Windows/Linux, 4GB RAM, 1 GB or more, 300 MB free space, processor speed 1.5 Ghz or faster	Open Source
Frontend	Camunda BPM 7.6.0 CMMN 1.1	OpenLayers 4.4.2 Angular JS 1.6.0	2-4 CPU, 4-16 GB RAM	Open Source
Backend	Geoserver 2.12.0	-	4 CPU, 8 GB RAM	Open Source
Backend	WSO2 ESB 5.0	-	1 CPU, 4 GB RAM	Open Source
Backend	SpringBoot 4.0	-	1 CPU, 2 GB RAM	Open Source
Backend	Ogr2ogr/GDAL 1.7.2	-	1 CPU, 2 GB RAM	Open Source
Database	PostgreSQL 10.0	Postgis 2.2.6	4 CPU, 8 GB RAM	Open Source

Table 3 List of layers of development and application frameworks

AngularJS Frontend

AngularJS is an open-source Front-end JavaScript framework. Its goal is to augment browser-based applications with Model–View–Controller (MVC) capability and reduce the amount of JavaScript needed to make web applications functional. These types of apps are also known as Single-Page Applications.

OpenLayers AddOn

OpenLayers API is built on top of Google Closure library. The library uses an OO approach so you will be getting used to OpenLayers in-built types like *ol.Map*, *ol.Collection*, *ol.layer.Vector*, *ol.source.KML*, *ol.Feature*, etc. In order to operate with object properties, we can use generic getters and setters.

Angular integrates seamlessly with OpenLayers. There are two ways of interaction:

- **Map to View**—this will be handled using the *onFeatureSelected* event handler
- **View to Map**—in our case we are registering to a message bus to capture search filter changes and update the map;

Camunda Process Engine

The core of Camunda BPM is an execution engine for BPMN, CMMN and DMN.
It is lightweight and requires less than 3MB of disk space. It can run in any Java Virtual Machine (JVM) and comes with extended integration for different runtime containers.

Java Spring Service Layer

To code and expose as services the interaction with the database we have implemented 2 Spring service layers: D-Layer and C-Layer.

5.1 Presentation and Content Layer

The Client Tier represents the point at which data is consumed by the system's users which include online users as well as external systems.

A standard Internet Browser is the primary client for the online Screening application. HTML pages are delivered to the client browser by ARIS application upon a user request. The Web Pages also include JavaScript functions. If JavaScript is turned-off, server-side validations are performed to ensure all validations are met.

OpenLayers makes it easy to put a dynamic map in any web page. It can display map tiles, vector data and markers loaded from any source. OpenLayers has been developed to further the use of geographic information of all kinds. OpenLayers has long been the standard choice for embedding a browsable map view into a webpage. A mature and comprehensive library (over 400k of minimised JavaScript), it has a moderate learning curve but is capable of many applications beyond a simple "slippy map": its features include full projection support, vector drawing, overview maps, and much more.

5.1.1 Balancing

These days the Internet is so widespread and ubiquitous it's hard to imagine it wasn't exactly there, as we know it, a decade ago. It has greatly evolved, from simple HTML producing clickable text, based on NCSA and then on Apache web servers, to an always-on communication medium used by more than 2 billion users worldwide. With the proliferation of permanently connected PCs, mobile devices and recently tablets, the Internet landscape is rapidly changing and entire economies have become digitally wired. Online services have become much more elaborate with a clear bias towards instantly available live information and entertainment. Security aspects of running online business have also significantly changed. Accordingly, websites are now much more complex than before, and generally require a lot more engineering efforts to be robust and scalable.

Traditional process or thread-based models of handling concurrent connections involve handling each connection with a separate process or thread, and blocking on network or input/output operations. Depending on the application, it can be very inefficient in terms of memory and CPU consumption. Spawning a separate process or thread requires preparation of a new runtime environment, including allocation of heap and stack memory, and the creation of a new execution context. Additional CPU time is also spent creating these items, which can eventually lead to poor performance due to thread thrashing on excessive context switching. All of these complications manifest themselves in older web server architectures like Apache's. This is a tradeoff between offering a rich set of generally applicable features and optimized usage of server resources.

From the very beginning, nginx was meant to be a specialized tool to achieve more performance, density and economical use of server resources while enabling dynamic growth of a website, so it has followed a different model. It was actually inspired by the ongoing development of advanced event-based mechanisms in a variety of operating systems. What resulted is a modular, event-driven, asynchronous, single-threaded, non-blocking architecture which became the foundation of nginx code.

nginx uses multiplexing and event notifications heavily, and dedicates specific tasks to separate processes. Connections are processed in a highly efficient run-loop in a limited number of single-threaded processes called workers. Within each worker nginx can handle many thousands of concurrent connections and requests per second.

5.2 Business Objects Layer

The Business layer will implement the business rules for the application. It will host the business service components as well as business objects (BO). These Business Services include Enterprise Java Beans and the BO's include the dependent JAVA classes that will provide service API's to the business rules and operations required by the application. Business components are software units, and process business logic.

The business components will implement the following:

- Business rules, such as calculations and validations
- Interfaces between the user interface and the resource layer

The business logic layer will run under the "Application Server" environment. Application Servers provide support for transaction control, thread management and other run-time services that make application development much simpler and more reliable.

5.2.1 Camunda BPM/CMMN

ARIS Backend Application is build on Camunda BPM, a Java-based framework. The main components are written in Java and we have a general focus on providing Java developers with the tools they need for designing, implementing and running business processes and workflows on the JVM. Nevertheless, we also want to make the process engine technology available to non-Java developers. This is why Camunda BPM also provides a REST API which allows you to build applications connecting to a remote process engine.
Camunda BPM can be used both as a standalone process engine server or embedded inside custom Java applications. The embeddability requirement is at the heart of many architectural decisions within Camunda BPM. For instance, we work hard to make the process engine component a lightweight component with as little dependencies on third-party libraries as possible. Furthermore, the embeddability motivates programming model choices such as the capabilities of the process engine to participate in Spring Managed or JTA transactions and the threading model.

In order to provide scale-up or fail-over capabilities, the process engine will be distributed to different nodes in a two node cluster. Each process engine instance will connect to the shared database.

CMMN allows modeling Cases. A case allows humans to do work in a more or less structured way in order to achieve something. Classic examples where case management is applied are Credit Application, Customer Support Management, Document Management, and so on.

The Camunda BPM web applications are based on a RESTful architecture.
Frameworks used:

- JAX-RS based Rest API
- AngularJS

- RequireJS
- jQuery
- Twitter Bootstrap

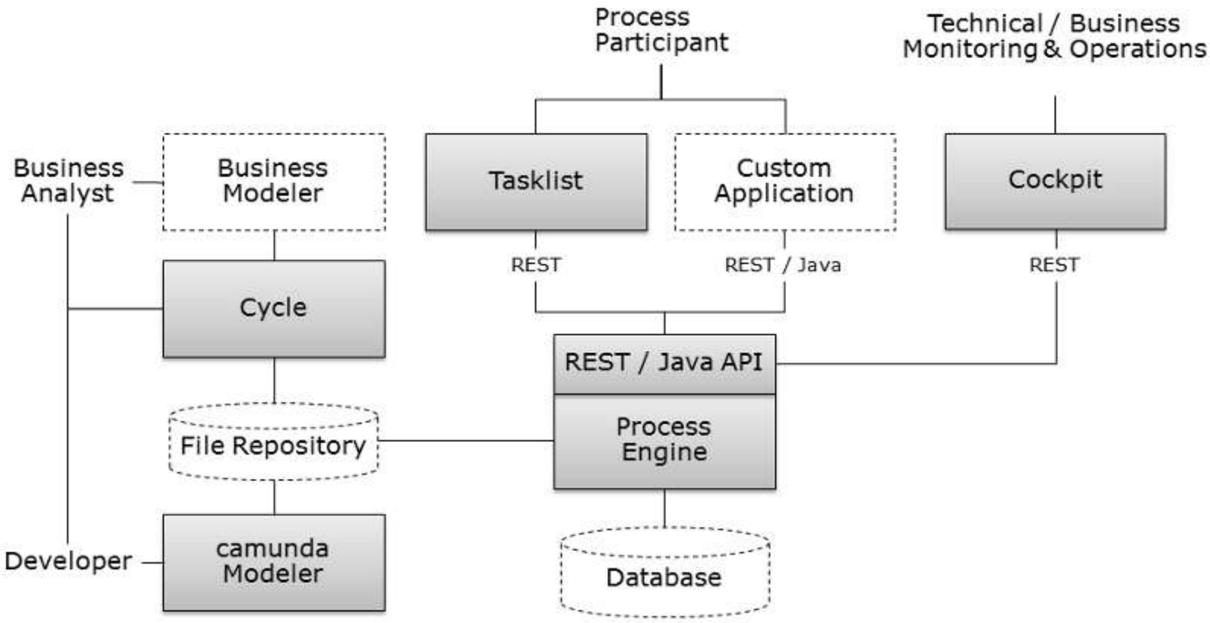

Figure 2 Camunda architecture Overview

Case instance refers to an instance of the case plan model. More specific, it is an instance of the single top-level stage in a case definition. The lifecycle of a case instance is the following:

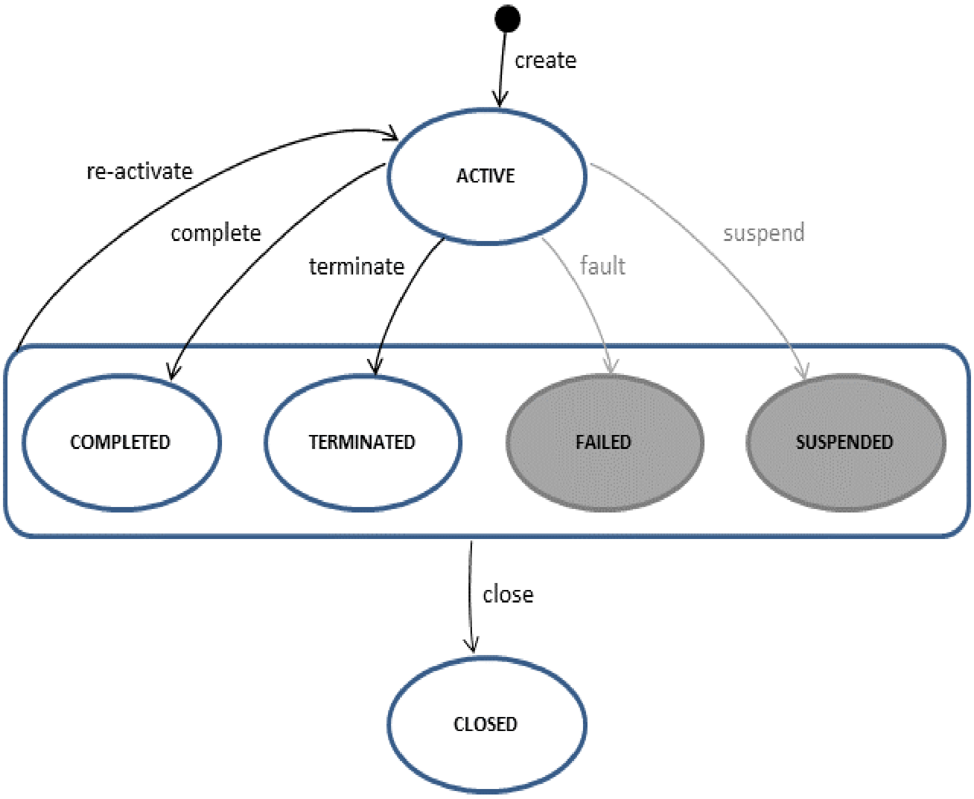

Figure 3 The lifecycle of a case instance

State	Description
active	The state active is the initial state when a case is instantiated via the CaseService API. Subsequently, all plan items defined in the case plan model are created and enter the state available.
completed	The transition complete automatically triggers when all plan items contained in the case plan model are completed, terminated, or disabled. With automatic completion enabled, only required plan items have to reach theses states. Furthermore, it is possible to manually complete a case instance via the CaseService API if it has no active tasks or stages and all required plan items are either completed, terminated, or disabled.
terminated	The transition terminate automatically triggers when the case instance's exit criteria are fulfilled.
closed	A case instance can be manually closed at any time via the CaseService API. This removes the case instance from the runtime database.

Table 4 The case states description

5.2.2 Geoserver

GeoServer is an open source server for sharing geospatial data. OGC compliant implementation of a number of open standards such as Web Feature Service (WFS), Web Map Service (WMS), and Web Coverage Service (WCS). Additional formats and publication options are available including Web Map Tile Service (WMTS) and extensions for Catalogue Service (CSW) and Web Processing Service (WPS).

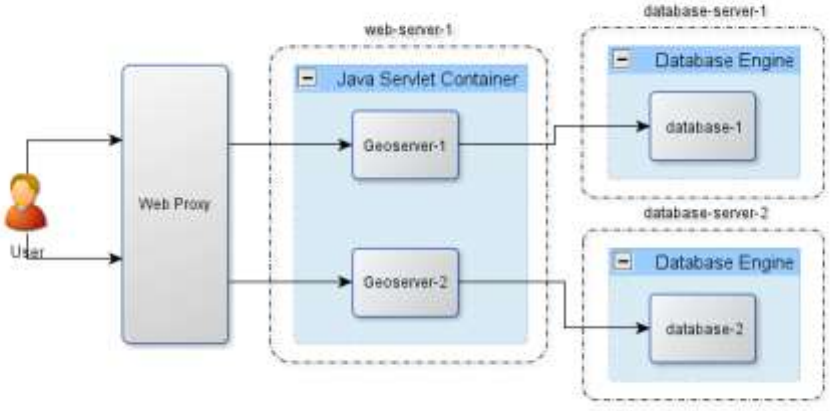

Figure 4 Geoserver Architecture Overview

5.3 Data Access Layer

Data Access Objects using Java Database Connectivity (JDBC) will manage the interface to the database. Persistence can be complex in large applications using protocols like JDBC. Neither the client nor the business component needs to be aware of this complexity. Moreover there are many forms of storage from databases, to flat files. Decoupling the persistence logic from the business components and client allows for a flexible, easy to maintain application. The Data Access Object (DAO) pattern allows for the abstraction of the persistence from the business component. The Data Access Object manages the connection to the data source to obtain and store data. It encapsulates all access to the data store.

5.3.1 Interoperability

WSO2 ESB allows interaction with external information systems which will be realized through interoperability framework platform MConnect. ARIS will use the interoperability platform MConnect as an interaction mechanism with IT systems of other Moldovan authorities.

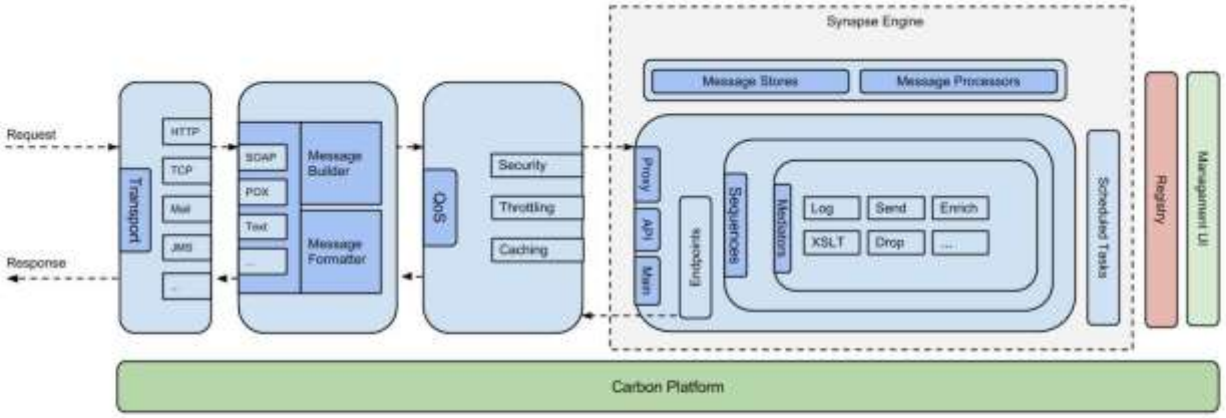

Figure 5 ESB Component Architecture

1. An application sends a message to the ESB.
2. The message is picked up by a transport.
3. The transport sends the message through a message pipe, which handles quality of service aspects like Security and Reliable Messaging. Internally, this pipe is the in-flow and out-flow of Axis2. The ESB can operate in two modes:
   - Message Mediation - A single pipe is used.
   - Proxy Services - Separate pipes connecting the transport to different proxy services are used.
4. Both message transformation and routing can be seen as a single unit. As the diagram specifies, there is no clear separation between message transformation components and routing components. WSO2 ESB call this the mediation framework. Some transformations happen before routing decision has taken. Some transformations happen after the routing decision has taken. This part is the Synapse implementation.
5. After this message is injected to the separate pipes depending on the destinations. Here again quality of service aspects of the messages is determined.
6. The transport layer takes care of the transport protocol transformations required by the ESB.
7. The diagram shows how a request propagates to its actual endpoint through the ESB using its architecture. Response handling is the reverse of this operation. There are other areas like Tasks and Events that are not shown in the diagram. All these components can be managed and monitored through WSO2 ESB management console.

5.4 Resource Layer

The resource layer includes the underlying resources that the application uses to deliver its functionality. This includes using PostgreSQL/Postgis Database and file system to persist information.

5.5 Database Architecture

ARIS will use PostgreSQL latest version (v.10 at current moment) and support DBMS upgrades to the latest version available in the future.
Information and data that need to be stored in Screening Relational database will be determined based on discussions with FSSA Stakeholders and IT staff and also during Design and Development phases of the project.

Three databases will be used:

Database	Role
ARIS_DB	Process database for ARIS
ARIS_BACKEND	ARIS case management artifacts for business processes
ARIS_GEOGIG	Holds informations for Geoserver GeoGig extension for versioning objects
ARIS_GEOSERVER	Hold Geoserver objects

Table 5 List of used databases

5.5.1 Data Model

Data Model is a method for describing data structures and a set of operations used to manipulate and validate that data. Data Model for ARIS application is as shown below:

ARIS database

ARIS process database model is described in attached document: **Address Register Information System – DBModel.docx**

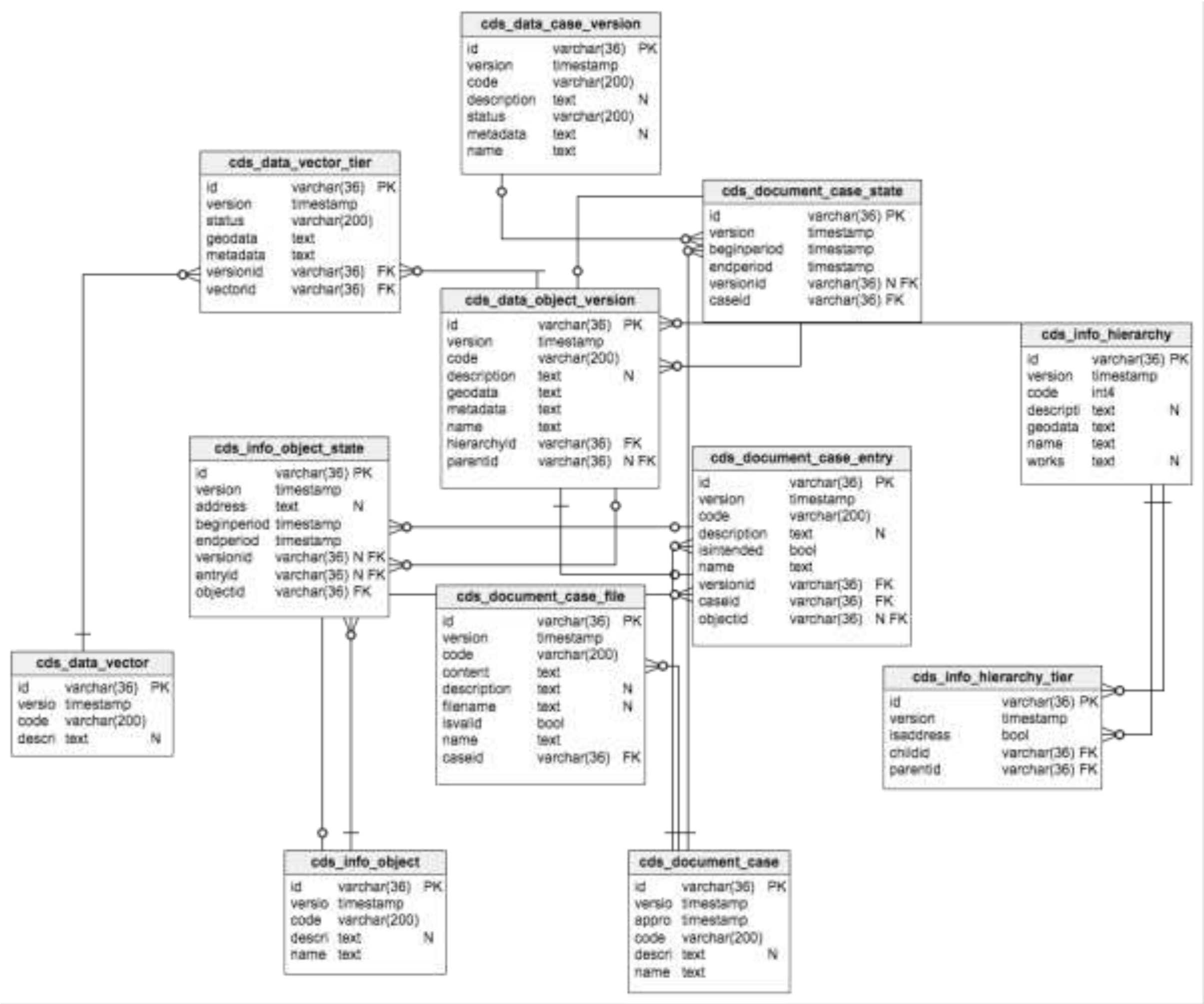

Figure 6 Database model

6 Deployment view

Our solutions and technologies are compatible with virtually any web server platform running on Microsoft Windows, MacOS, Linux, and UNIX.

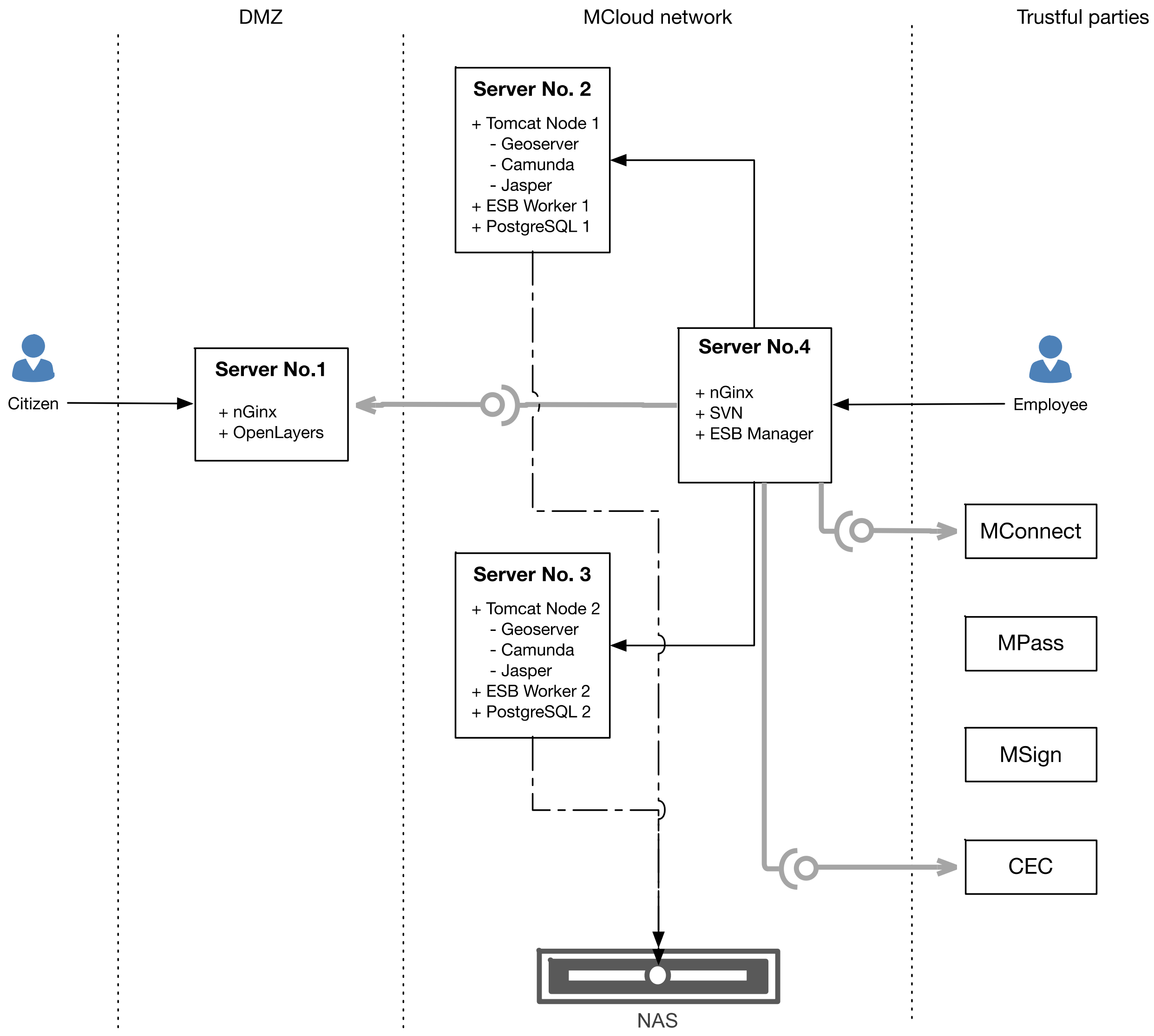

Figure 7 Deployment Diagram

ARIS Frontend and Backend applications:

Server No	HDD (GB)	Processor (No)	Memory (GB)
1	50	1	4
2	100	2	8
3	100	2	8

Server No	HDD (GB)	Processor (No)	Memory (GB)
4	50	1	4

Table 6 Servers list

Network Attached Server (NAS): Capacity: 1 TB. It's recommended to attach it to Servers No1 and No. 2 where database is located, used as postgreSQL data folder to prevent disk full when database is increasing.

6.1 Classes, interfaces, artefacts

The documents related to the objectives such as classes, interfaces, artefacts are attached in JavaDOC (html) format in archives:

- dlayer-jdoc.zip
- processes-jdoc.zip

which are part of this SDD.

7 Software Requirements Traceability Matrix

ID	Name	Description	Priority	Type	ToR - type
MIGR 009	The objective of the Data migration process is to	The objective of the Data migration process is to migrate, convert and test all existing data that is necessary for testing and for the operation of the new application.	1	non-functional	Data migration
MIGR 010	The Contractor shall develop a Migration Strategy	The Contractor shall develop a Migration Strategy which identifies sources and targets of the data migration and describes an overall approach to migrate, convert and test all existing data. The Migration Strategy shall address all important issues, e.g.: // checking and testing data migration; migrating of historical data; inconsistency among data import; // managing fractional, corrupt, etc. data records; converting of the classifiers in new system; // indicating status of the migrated data to distinguish them from new records.	1	non-functional	Data migration
MIGR 011	The developer shall study the existing data models	The developer shall study the existing data models in all systems from which data shall be migrated and the new data model and shall propose a detailed design of module for data migration.	1	non-functional	Data migration
MIGR 012	The Contractor shall develop a Migration Plan	The Contractor shall develop a Migration Plan which describes the method of the data migration, identifies software tools and defines and schedules stages and activities of the migration.	1	non-functional	Data migration
MIGR 013	The Contractor shall develop the necessary software	The Contractor shall develop the necessary software tools, e.g. scripts to implement the migration. These tools provides the following functionality: // transfer data from existing system to ARIS; // checks quality (consistency, completeness, etc.) of the migrated data; // create statistical reports – number of objects before and after migration; control numbers, areas; // stores data quality attributes at ARIS; create reports; // the readiness status of each migration process is traceable by operators.	1	non-functional	Data migration
MIGR 014	The Contractor shall elaborate manuals for IT prof	The Contractor shall elaborate manuals for IT professionals regarding the use of migration tools.	1	non-functional	Data migration
MIGR 015	The Contractor shall train IT professionals regard	The Contractor shall train IT professionals regarding the approach to migration and use of migration software tools.	1	non-functional	Data migration

ID	Name	Description	Priority	Type	ToR - type
FR 15.10.	The logged event will save the following categories	The logged event will save the following categories of data (depending on the type of the logged event: // identifier of the user who generated the event; category of the logged event; // timestamp of event logging; // resource of the IT application that generated the business event; record affected by the business event; // action performed by the user.	1	functional	functional
FR 15.11.	The system shall keep logs for at least 6 months a	The system shall keep logs for at least 6 months and then archive it.	1	functional	functional
FR 15.12.	The IT System will deliver a mechanism to generate	The IT System will deliver a mechanism to generate reports related to logged events.	1	functional	functional
TGEN 001	The system interface shall be Multilingual (Romani	The system interface shall be Multilingual (Romanian, Russian and English) with mechanism to add new language without reprogramming, through attachable resource files.	1	functional	general system req
TGEN 002	The content of the database will be inputted in Ro	The content of the database will be inputted in Romanian, Russian and English.	1	functional	general system req
TGEN 003	The user's Interface shall be optimized to 1360x76	The user's Interface shall be optimized to minimum 1360x768 resolution up to 1920x1080 resolution.	1	functional	general system req
TGEN 004	ARIS shall have the possibility to adjust the user	ARIS shall have the possibility to adjust the user's interface (shall deliver a responsive interface) depending on the device used (notebook, netbook, desktop PC, tablet).	1	functional	general system req
TGEN 005	The system offers an accessible and intuitive inte	The system offers an accessible and intuitive interface to human users.	1	functional	general system req
TGEN 006	The user's Interface elements shall comply with Le	The user's Interface elements shall comply with Level A of Web Content Accessibility Guidelines (WCAG) 2.0.	1	functional	general system req
TGEN 007	ARIS will ensure functionalities necessary to use	ARIS will ensure functionalities necessary to use the digital signature and mobile signature for all categories of actors.	1	functional	general system req
TGEN 008	The IT System shall have integrated functions for	The IT System shall have integrated functions for searching and filtering after Metadata files/documents, profiles of mobilization resources or of authorized users (search records, documents, notifications, acts, etc.), search by the calendar, search by business events of mobilization processes, etc.). // The procedures of information and records retrieval shall be performed via simple search (specification of search series) or via more complex search forms allowing filtering the information (QBE forms). Regardless of the type of searched information, the user shall utilize the same method of queries and retrieval of information for any section of the software.	1	functional	general system req

ID	Name	Description	Priority	Type	ToR - type
TGEN 009	In addition to the searching module implemented ba	In addition to the searching module implemented based on QBE principle, which would offer the possibility to define visually sophisticated queries, the user interface shall offer the possibility to refine the search results by ensuring the possibility to filter the data in the list containing the search results.	1	functional	general system req
TGEN 010	The IT system user interface shall ensure filterin	The IT system user interface shall ensure filtering the records that match the search criterion presented by users depending on their rights of access.	1	functional	general system req
TGEN 011	The content of any table with search results shall	The content of any table with search results shall have the possibility to be exported in any of the following format: XLS, CSV and PDF.	1	functional	general system req
TGEN 012	ARIS shall offer API interfaces to interact with e	ARIS shall offer API interfaces to interact with external IT systems.	1	functional	general system req
TGEN 013	ARIS will ensure compatibility with W3C XForms sta	ARIS will ensure compatibility with W3C XForms standard.	1	functional	general system req
TGEN 014	The IT system shall be optimized in the minimum da	The IT system shall be optimized in the minimum data transfer between the client computer and server (e.g. implement AJAX with JSON), having focused on avoiding the redundant requests as much as possible.	1	functional	general system req
TGEN 015	ARIS shall have at its basis at least a three-leve	ARIS shall have at its basis at least a three-level architecture (with a distinct level for data) based on SOA.	1	functional	general system req
TGEN 016	The ARIS potentially variable information (paramet	The ARIS potentially variable information (parameters, ways if data storage, ways of connection with external services, etc.) shall be configurable and would not require solution recompilation or direct interventions into the DB.	1	functional	general system req
TGEN 017	The IT System shall use open standards for formats	The IT System shall use open standards for formats and communication protocols.	1	functional	general system req
TGEN 018	The services exposed to the public by ARIS shall b	The services exposed to the public by ARIS shall be technologically neutral (Operation System, Internet Explorer, etc.).	1	functional	general system req
PERF 001	The average server reply time shall not exceed 3 s	The average server reply time shall not exceed 3 seconds at system load indicated in (PERF 002, PERF 003). Replay time shell be logged for several time-consuming events. List of such events shell be elaborated at design stage with beneficiary.	1	performance	performance
PERF 002	The system must be capable to allow activity of ov	The system must be capable to allow activity of over 1100 authorized users of category Administrator, Level 1 LPA Operator, Level 2 LPA Operator, Address Validator, Street Validator and Classificatory Administrator. Number of service consumers and appropriate load for each consumer will be defined at analytical and design stage with beneficiary.	1	performance	performance

ID	Name	Description	Priority	Type	ToR - type
PERF 003	The System shall enable the competing activity of	The System shall enable the competing activity of at least 200 concurrent users and servicing of at least 300 simultaneous queries with constant response time indicated in PERF 001.	1	performance	performance
PERF 004	Data collection of the information system will hos	Data collection of the information system will host more than 2 million of addresses and will perform annually more than 20 000 transactions of addresses update.	1	performance	performance
PERF 005	Prior to the delivery of IT solution, ARIS perform	Prior to the delivery of IT solution, ARIS performance test shall occur.	1	performance	performance
PERF 006	Performance testing shall include at least two com	Performance testing shall include at least two components: system load testing and system stress testing. Test cases shell be elaborated by contractor in coordination with beneficiary. Test system shell be loaded with data conform PERF 004.	1	performance	performance
SR 001	The IT System guarantees full storage and integrit	The IT System guarantees full storage and integrity of ARIS DB content.	1	non-functional	security and protection
SR 002	Public information is made available to anonymous	Public information is made available to anonymous users.	1	non-functional	security and protection
SR 003	Access to functions granted to unauthorized and no	Access to functions granted to unauthorized and non-authenticated users shall be monitored using protection means against overstressing the service by one or several network hubs.	1	non-functional	security and protection
SR 004	The Security subsystem shall provide functionality	The Security subsystem shall provide functionality for single sign on, users' rights, password registry, etc. for all users of the system. The sub-system is also available for all other subsystems to check in the system for authorization rights.	1	non-functional	security and protection
SR 005	ARIS shall include a comprehensive security framew	ARIS shall include a comprehensive security framework	1	non-functional	security and protection
SR 006	ARIS shall include security related data transmiss	ARIS shall include security related data transmission, including: Service endpoint (respondent) authentication; // Client principal (initiator) authentication; Message integrity; // Message confidentiality; Replay detection.	1	non-functional	security and protection
SR 007	ARIS shall adopt means which will make possible th	ARIS shall adopt means which will make possible the encryption of data in database, messages and communication channels.	1	non-functional	security and protection
SR 008	ARIS shall provide monitoring functions based on u	ARIS shall provide monitoring functions based on unified methods to monitor user interactions regarding use of services and data manipulation.	1	non-functional	security and protection
SR 009	ARIS shall ensure a regular review of user informa	ARIS shall ensure a regular review of user information on access. At least every 6 months, Security unit reviews information on access in order to disclose any unauthorized access or data leaking.	1	non-functional	security and protection

ID	Name	Description	Priority	Type	ToR - type
SR 010	Access to editing of information objects and gener	Access to editing of information objects and generation of documents from the system shall be limited by the objects identified in the application submitted by client. // This should not limit viewing access of different objects.	1	non-functional	security and protection
SR 011	All users (including end-users, administrators, de	All users (including end-users, administrators, developers) shall have a unique identifier (user ID), which must not contain signs of user access level.	1	non-functional	security and protection
SR 012	The user ID administration shall include: recognit	The user ID administration shall include: recognition of each user; // the authentication of each user; // obtain authorization from the responsible manager to issue of the user ID; // ensuring that the user ID is issued specifically to a certain person; cancelling user account after a specified time period of inactivity // (idle for no more than 2 months); // implementation of backup copies of user Ids; // setting the organizational structure of users. (Creating of organizational structure: the formation of lists of users and roles, departments and organizations; // access control features (Setting permissions).	1	non-functional	security and protection
SR 013	Before granting access to the system, users should	Before granting access to the system, users should be informed that the use of information (especially personal data) is monitored and that their unauthorized use can be prosecuted in accordance with applicable law.	1	non-functional	security and protection
SR 014	All users of ARIS are responsible for their ID's a	All users of ARIS are responsible for their ID's and passwords: users can choose and change their own passwords; // users are unable to access account after 5 incorrect authentication attempts; // previous user passwords are stored and re-use is prevented; passwords are not visible on the screen; // passwords are stored in encrypted form, using one-way encryption algorithm (function hash).	1	non-functional	security and protection
SR 015	ARIS shall include a mechanism for restoring lost	ARIS shall include a mechanism for restoring lost passwords.	1	non-functional	security and protection
SR 016	ARIS working session regarding registers and perso	ARIS working session regarding registers and personal data shall be logout automatically after more than 15 minutes of user inactivity, which prevents any further access until the user unlocks the session by repeating the procedure of identification and authentication.	1	non-functional	security and protection
SR 017	All fields of forms filled in by users must be val	All fields of forms filled in by users must be validated by type of both the client and server.	1	non-functional	security and protection
SR 018	When the system communicates with other systems di	When the system communicates with other systems digital certificates shall be used for identity.	1	non-functional	security and protection

ID	Name	Description	Priority	Type	ToR - type
SR 019	For sensible transactions, immediately after their	For sensible transactions, immediately after their execution, will be used the time stamping service.	1	non-functional	security and protection
SR 020	The System shall be secured against OWASP Top 10 v	The System shall be secured against OWASP Top 10 vulnerabilities.	1	non-functional	security and protection
SR 021	The System shall ensure confidentiality of data tr	The System shall ensure confidentiality of data transmitted-received via communications channels. System data exchange is done only via secure channels.	1	non-functional	security and protection
SR 022	Access to the ARIS shall be monitored.	Access to the ARIS shall be monitored.	1	non-functional	security and protection
SR 023	Interaction with IT Systems shall be performed thr	Interaction with IT Systems shall be performed through an authentication procedure using the digital certificate.	1	non-functional	security and protection
SR 024	Access to functions for non-anonymous users shall	Access to functions for non-anonymous users shall be granted by their authentication, using user + Password or digital certificate (via MPass).	1	non-functional	security and protection
SR 025	The System will deliver strong mechanisms to secur	The System will deliver strong mechanisms to secure the procedure safety for users' authentication and authorization.	1	non-functional	security and protection
SR 026	All users' actions shall be recorded into electron	All users' actions shall be recorded into electronic logs.	1	non-functional	security and protection
SR 027	The System shall make a periodic signal that tells	The System shall make a periodic signal that tells about its functional status.	1	non-functional	security and protection
SHC 001	The System shall have the possibility to be instal	The System shall have the possibility to be installed on both dedicated servers and on virtual solutions.	1	constraint	software, hardware and communications
SHC 002	The system architecture should be in line with the	The system architecture should be in line with the Cloud First strategy promoted by the e-Government Center.	1	constraint	software, hardware and communications
SHC 003	It is necessary to demonstrate the capacity of vir	It is necessary to demonstrate the capacity of virtualization via the delivery of a system image to the Beneficiary that could be uploaded and become operational with minimum configurations on one of the virtualization solutions available on the market.	1	constraint	software, hardware and communications
SHC 004	The System shall be accessed through communication	The System shall be accessed through communication channels of at least 128kbps.	1	constraint	software, hardware and communications
SHC 005	The system shall be implemented using open source	The system shall be implemented using open source license to the maximum possible extent,	1	constraint	software, hardware and communications
SHC 006	The Developer shall state explicitly in the offer	The Developer shall state explicitly in the offer the software platform based on which the ARIS has been built and the software platform required for the ARIS functioning.	1	constraint	software, hardware and communications

ID	Name	Description	Priority	Type	ToR - type
SHC 007	If the software platform used to develop and opera	If the software platform used to develop and operate the ARIS is based on commercial IT solutions, requiring license procurement, the Developer will include in the price offer the delivery of all licenses required for ARIS development and operation (the Developer shall purchase on behalf of the // S.E. „Cadastru" all licenses required for the development and operation of the IT System).	1	constraint	software, hardware and communications
SHC 008	If the software platform used to develop and opera	If the software platform used to develop and operate the ARIS is based on commercial IT solutions, requiring license procurement, the Developer will include in the price offer the overall amount charged for licensing when: // doubling the number of users; // doubling the number of processing units (CPU or CPU kernels); doubling the number of server application hubs/DB.	1	constraint	software, hardware and communications
SHC 009	All software must be provided with unlimited durat	All software must be provided with unlimited duration license(s), allowing upgrading to new versions of third party products during warranty period.	1	constraint	software, hardware and communications
SHC 010	All software must be provided with an unlimited nu	All software must be provided with an unlimited number of concurrent users (e.g. Web based)	1	constraint	software, hardware and communications
SHC 011	All software must be provided for an unlimited num	All software must be provided for an unlimited number of users for any infrastructure component (i.e. application server, plug-in, etc.) needed for full operation.	1	constraint	software, hardware and communications
SHC 012	The system shall have a modular architecture, whic	The system shall have a modular architecture, which shall follow n-layer architectural pattern with clear separation between layers. System components shall be loosely coupled and have clear communication interfaces	1	constraint	software, hardware and communications
SHC 013	The system shall expose its functionality as API t	The system shall expose its functionality as API through Web Services. The API shall be clearly and comprehensively documented.	1	constraint	software, hardware and communications
SHC 014	The system implements public available parts of th	The system implements public available parts of the service as Web Parts to be integrated in Government Portal.	1	constraint	software, hardware and communications
SHC 015	The system shall be logically decoupled through ab	The system shall be logically decoupled through abstract interfaces from modules implementing functionalities such as logging, notifications, authentication, scheduling if such modules are in scope of current system. These modules will work also with MCloud shared platform level modules.	1	constraint	software, hardware and communications
SHC 016	The system must be highly configurable and shall n	The system must be highly configurable and shall not be tied in any way to specific physical resources, such as locations on disks or types of devices. The configuration shall allow changes of important parameters preferably without the need to restart the running system	1	constraint	software, hardware and communications

ID	Name	Description	Priority	Type	ToR - type
SHC 017	The Contractor must license all software to the Be	The Contractor must license all software to the Beneficiary allowing designated personnel to perform diagnostics, installation, update / upgrade and repair / debug activities without any external assistance. Beneficiary should receive support for at least for three years from Contractor in any part of the system.	1	constraint	software, hardware and communications
SHC 018	The System shall be tolerant to errors by offering	The System shall be tolerant to errors by offering support for clustering and fail over for the whole platform and own components.	1	constraint	software, hardware and communications
SHC 019	It is advisable to ensure that the service parts e	It is advisable to ensure that the service parts exposed to the public are technologically neutral.	1	constraint	software, hardware and communications
SHC 020	The WEB Explorer is the recommended generic softwa	The WEB Explorer is the recommended generic software for the operation and interaction with ARIS.	1	constraint	software, hardware and communications
SHC 021	The System shall be compatible with at least 2 the	The System shall be compatible with at least 2 the most recent versions of the following WEB browsers: Microsoft Internet Explorer, Mozilla Firefox, Google Chrome, Safari and Opera.	1	constraint	software, hardware and communications
SHC 022	Compatibility with Microsoft Internet Explorer is	Compatibility with Microsoft Internet Explorer is mandatory.	1	constraint	software, hardware and communications
SHC 023	ARIS shall incorporate a Heart-beat service to per	ARIS shall incorporate a Heart-beat service to periodically communicate the system normal work status.	1	constraint	software, hardware and communications
SHC 024	The System shall include configurable means for te	The System shall include configurable means for technical logging.	1	constraint	software, hardware and communications
SHC 025	The System shall be able to produce at least the f	The System shall be able to produce at least the following levels of technical logging: info; warning; critic; error.	1	constraint	software, hardware and communications
SHC 026	The Developer shall list the means to be used for	The Developer shall list the means to be used for system troubleshooting.	1	constraint	software, hardware and communications
SHC 027	The Developer shall prepare means that facilitate	The Developer shall prepare means that facilitate the system administration functions: // starting the system components; // stopping the system components; restarting the system components, creating a DB back-up, // recovery of data using the indicated back-up, refreshing the system operational memory.	1	constraint	software, hardware and communications
SHC 028	The System shall operate in TCP/IP networks and, e	The System shall operate in TCP/IP networks and, especially, in HTTPS.	1	constraint	software, hardware and communications
SHC 029	The Developer shall suggest other network services	The Developer shall suggest other network services and utilities necessary for system operation.	1	constraint	software, hardware and communications
SDI 001	SDI of ARIS must support ISO19139:2007 metadata da	SDI of ARIS must support ISO19139:2007 metadata dataset collections profile compatible with the INSPIRE Metadata Implementing Rules guidance based on ISO 19115 / ISO 19119.	1	functional	Spatial Data Infrastructure requirements
SDI 002	SDI of ARIS must support INSPIRE Data Specificatio	SDI of ARIS must support INSPIRE Data Specifications on Address Data Model	1	functional	Spatial Data Infrastructure requirements

ID	Name	Description	Priority	Type	ToR - type
SDI 003	SDI of ARIS must support INSPIRE Download services	SDI of ARIS must support INSPIRE Download services, View services, Discovery Services, Transformation services.	1	functional	Spatial Data Infrastructure requirements
SDI 004	ARIS must support SDI services, as defined by the	ARIS must support SDI services, as defined by the OGC Standards with support for minimum: // Web Map Service (WMS) v1.1.1 (or later); Web Feature Service (WFS) v1.1.0 (or later); Web Coverage Service (WCS) v1.0.0 (or later); // Catalog Service Web Profile (CSW) v2.0.0 (or later).	1	functional	Spatial Data Infrastructure requirements
SDI 005	ARIS must provide a password-protected access to W	ARIS must provide a password-protected access to WMS and WFS for desktop-based or web-based GIS software clients.	1	functional	Spatial Data Infrastructure requirements
SDI 006	ARIS must support SDI services web-based managemen	ARIS must support SDI services web-based management interface with min support for: // Setting up different levels of access for different categories of users for CSW service; // On-line editing of metadata and uploading of metadata files in XML format for CSW service; // On-line metadata search and discovery by using an attribute (search) query interface and spatial query (web map search interface) for CSW service; // On-line set-up and configuration of WMS, WFS and WCS services; // On-line preview and retrieval of SDI datasets in the form of GIS data files (for WMS, WFS and WCS) with support for min SHP and GML as vector layer(s) file formats and min GeoTIF as raster layer(s) file formats.	1	functional	Spatial Data Infrastructure requirements
SDI 007	ARIS must support Web Map Publishing (WMP) with mi	ARIS must support Web Map Publishing (WMP) with min support for: managing and portrayal of SDI services and datasets with support // for standard web technologies (e.g. JavaScript, XML, etc.); // authentication and authorization services with user, group and service administration capabilities using web-based user interfaces; // web-based management interface for web mapping content and design of customized web mapping applications.	1	functional	Spatial Data Infrastructure requirements
SDI 008	SDI of ARIS must provide services for integration	SDI of ARIS must provide services for integration with other external systems throughout MConnect.	1	functional	Spatial Data Infrastructure requirements
DBMS 001	RDBMS must be delivered under open source license	RDBMS must be delivered under open source license (GNU Public License or similar), which must also apply to all database custom code (SQL scripts, types, functions, triggers, packages etc.) provided during supply delivery and created during customization and integration	1	functional	RDBMS requirements
DBMS 002	RDBMS must support remote connectivity options (co	RDBMS must support remote connectivity options (connectors, drivers, etc.) for direct read/write access by external desktop-based or web-based software client(s)	1	functional	RDBMS requirements
DBMS 003	Native 64-bit RDBMS version supporting native 64-b	Native 64-bit RDBMS version supporting native 64-bit Server OS environment must be installed	1	functional	RDBMS requirements

ID	Name	Description	Priority	Type	ToR - type
DBMS 004	RDBMS must support a full-scale automatic replicat	RDBMS must support a full-scale automatic replication	1	functional	RDBMS requirements
DBMS 005	RDBMS must support full-scale back up/recovery cap	RDBMS must support full-scale back up/recovery capabilities using standard SQL data format(s) as well within native internall functionality.	1	functional	RDBMS requirements
DBMS 006	RDBMS must support full-scale self-maintenance cap	RDBMS must support full-scale self-maintenance capabilities with min support for automatic re-indexing, clean-up of temporary records and recovery of storage space.	1	functional	RDBMS requirements
DBMS 007	RDBMS native GIS application extension compliant w	RDBMS native GIS application extension compliant with the OGC Simple Features Specifications for SQL must be installed (e.g. PostGIS)	1	functional	RDBMS requirements
DBMS 008	RDBMS and RDBMS native GIS application extension m	RDBMS and RDBMS native GIS application extension must provide a back- end support for the introduction and implementation of the SDI, supporting min CSW, WMS and WFS services.	1	functional	RDBMS requirements
DBMS 009	RDBMS must support clusterization with load balanc	RDBMS must support clusterization with load balancing.	1	functional	RDBMS requirements
DBMS 010	RDBMS must support internal audit capabilities to	RDBMS must support internal audit capabilities to manage and control users' actions and use of privileges.	1	functional	RDBMS requirements
DBMS 011	RDBMS must support data protection capabilities su	RDBMS must support data protection capabilities such as SSL and data encryption. In some cases data encryption equivalent shell be supplied by developer.	1	functional	RDBMS requirements
INI 001	ARIS will run on the MCloud platform	ARIS will run on the MCloud platform	1	constraint	IT related aspects and field-related initiatives
INI 002	The IT System will use the MPass service as a mech	The IT System will use the MPass service as a mechanism for users' authentication and authorization.	1	constraint	IT related aspects and field-related initiatives
INI 003	The IT System will use the MSign service as a mech	The IT System will use the MSign service as a mechanism for digital signature application and validation.	1	constraint	IT related aspects and field-related initiatives
INI 004	The IT System will use the MLog service as a mecha	The IT System will use the MLog service as a mechanism for logging for the critical business events.	1	constraint	IT related aspects and field-related initiatives
INI 005	The IT System will use the MNotify service as a me	The IT System will use the MNotify service as a mechanism for notifying the authorized users.	1	constraint	IT related aspects and field-related initiatives
INI 006	The IT System will use the interoperability platfo	The IT System will use the interoperability platform MConnect as an interaction mechanism with IT systems of other Moldovan authorities.	1	constraint	IT related aspects and field-related initiatives
INI 007	ARIS will use PKI services offered by the single t	ARIS will use PKI services offered by the single technological platform // MCloud.	1	constraint	IT related aspects and field-related initiatives
INI 008	ARIS will identify a set of at least 10 performanc	ARIS will identify a set of at least 10 performance indicators which will be automatically published by the single technological platform monitoring services.	1	constraint	IT related aspects and field-related initiatives

Table 7 Software Requirements Traceability Matrix