Modernization of Government Services
in the Republic of Moldova

Project ID No. P148537

Terms of Reference

CONSULTING SERVICES FOR THE DESIGN, DEVELOPMENT, CONFIGURATION AND DEPLOYMENT OF THE MDELIVERY INFORMATION SYSTEM

1. Background

The Government of Moldova is determined to fundamentally change the way public services are provided in Moldova through a variety of interventions for modernization of service delivery, which combat corruption, foster a customer care culture, enhance access, as well as increases efficiency in the Moldovan public administration. Therefore, one of the main objectives of the Administration Reform Strategy 2016-2020 is the modernization of public services.

The Government of Moldova, via e-Governance Agency (EGA) is implementing a World Bank-funded PAR operation, planned for 2018 – 2023 – Modernization of Government Services Project (MGSP)[1]. The e-Governance Agency is responsible for development and implementation of Government e-Transformation Agenda, which intends to ensure country's sustainable development through efficient use of ICT. EGA manages e-transformation activities at governmental level on behalf of the State Chancellery.

The Government of Moldova, in accordance with the Public Administration Reform (PAR) Strategy for the years 2016-2020 (especially the component "Modernization of Public Services") is undergoing a major transformation exercise (qualitative and quantitative) of administrative services, provided by central public administration authorities through: a) removing outdated public services or merging several services in one; b) increasing access to local public services through various channels; c) reducing the number of documents required for services, and the delivery time; e) ensuring a high level of satisfaction with the quality of government service delivery.

Typically, a public service has a result delivered to the user at the end of delivery process. Depending on the delivery channel, the result may be electronic requiring no shipping and pickup or the result may be a physical good (parcel) requiring shipping and pick-up. In case of public services resulting in physical goods (e.g. passports, IDs, certificates, extracts, permits, driving licenses etc.), currently citizens or business are usually required to pick-up these goods themselves from the Public Service Providers offices.

[1] Modernization of Government Services in the Republic of Moldova
http://projects.worldbank.org/P148537?lang=en

Nevertheless, physical movement of goods introduces a huge number of concerns, such as that shippers cannot negotiate a better price for delivery of parcels; deliverers have difficulties publicizing the service to a large enough audience; receivers have few options to get deliveries and instead have to waste time to get the goods themselves; physical goods have very different properties each possibly incurring additional complexity regarding pick up, storage, handling, preservation and movement; every participant has a hard time handling wrong deliveries, returns, refunds etc.

Due to the fact that nowadays users have become accustomed to new means of service delivery, expecting the same level of variety of services from the public sector, Users want their interactions to be convenient, and they prefer to be online rather than inline.

Delivery of parcels is an integral part of a digital society. Therefore, MDelivery will introduce a unified and integrated delivery mechanism (standard integration point between delivery actors) in order to improve the logistic capability of Public Service Providers to deliver, sort and track physical goods (as results of public services) to their users such as individuals and legal entities.

MDelivery will be a full-fledged solution to provide a reliable, flexible and efficient mechanism for delivery, using various delivery channels and options (same-day, next business day or other time, express, priority, international), parcel handling, sorting and tracking where the physical parcel is marked and labeled accordingly during delivery. MDelivery will focus only on shipment of non-hazardous, non-perishable, regular size and weight parcels thus avoiding dealing with constraints regarding special handling, storage conditions and personnel qualifications.

MDelivery will consist of a single payment approach where the Shipper integrates the shipment as a product in his electronic commerce cart. If the cart is paid in full the Shipper, then confirms the shipment to MDelivery for subsequent processing and delivery. This of course does not preclude shippers offering free delivery to end users by waiving shipment costs. The costs will still need to be disbursed to delivery services thus delivery costs still need to be listed and confirmed by the Shipper before shipment is authorized.

The following entities are interested or will be involved in the development and/or operation of MDelivery:

- e-Governance Agency, as owner of MDelivery, who is responsible for development, implementation, and operation of the information solution.
- Carriers, as providers of transportation of physical parcels with various delivery options (same-day, next business day or other time, express, priority, international).
- Public service providers (Public authorities and institutions of all levels) as shippers and owners of information systems who will integrate with MDelivery to deliver physical parcels to individuals and legal entities regarding various public services.
- Individuals and legal entities, as receivers of physical parcels from public service providers regarding various public services.

2. Objective of the Assignment

The Client is looking for an ICT consulting company to develop the MDelivery Information System with demonstrated experience in the design and implementation of similar complexity projects to perform key client-facing activities, and to provide on-going maintenance and technical support.

3. Scope of work and Development approach

The scope of work of this assignment is to design, develop, configure, and deploy the information system as a fully functional product with all functionalities in place, according to the specifications iteratively defined by the Client (the indicative set of requirements is listed in **Annex 1** and **Annex 2**) and following the development approach described below.

The development of the solution will follow agile iterative software development principles. Since there are many interpretations of agile software development and in order to avoid misunderstandings, this section provides key technology principles to be used in development of the solution.

Iterative development
In contrast to waterfall software development approach, the solution shall be developed in iterations named sprints. This means that the implementation of different functionalities will take place in phases with some modules being in production while others still being in development. The priorities of functionalities included in a sprint will be determined by the Client. Sprint duration will be determined by the Client together with the Consultant.

Agile development
The development shall follow agile principles by allowing change and flexibility in implementation. Client will maintain the master list of generic requirements for the solution– *product backlog*, which consists of ordered business and technical requirements as seen by the Client. Items in product backlog are ordered by the Client by their priorities. Client is free to manage the product backlog by adding new items to it, removing items and reordering them as he/she desires. At the beginning of each sprint, the topmost **N** items that fit into a sprint are taken, and a *sprint backlog* is built out of them. Items in sprint backlog are further detailed and distributed to developers. Sprint backlog is not changed during the sprint.

Working product in each iteration
Each sprint ends up in a working product which is presented to the Client for acceptance in the last day(s) of sprint. The working product shall meet the agreed criteria – Definition of Done (e.g. it must be fully functional, fully tested, accompanied with relevant unit tests, accompanied with relevant documentation where necessary, complete commented source code supplied etc.). Payments will be made upon successful delivery of working packages (one or more working products). In case the deliverables contain defects for reasons not imputable to the Client, the Consultant shall fix them without impacting the time schedule and at no additional costs, including possible visits to Client site. Working products from different sprints can be combined into a release deployed in production at Client's discretion. Any incidents reported by the Client after the release, shall be solved by the Consultant according to the agreed Service Level Agreements (SLAs) as defined in Annex 2, p.10 Support and Warranty requirements.

To ensure that the development team is in position to deliver on time working products, a Client representative – typically named *the Product Own*er in agile methodologies – is permanently available to the team for answering eventual questions, thus not slowing down the implementation pace.

The Consultant will appoint a Scrum Master from the team of key or non-key experts for the entire duration of the project.

The Scrum Master will be responsible for the day-to-day liaison with the Client; s/he must ensure the internal coordination and guidance of the project experts and the project coordination with external counterparts.

The Scrum Master must also ensure the availability of suitable experts in accordance with the project planning documentation.

Client involvement

In contrast with commonly used waterfall model for procurement and implementation of information systems for the government, the Client designated person – Product Owner – will be heavily involved in the development process. The Product Owner will have three core responsibilities:

1. Maintenance of product backlog – the owner will maintain the product backlog up to date, so it reflects prioritized list of desired functionalities.
2. Answering to questions coming from developers – the owner will be at all time available to the development team for answering their eventual clarification questions, thus avoiding complex and formal communication within the project. This is essential to ensure the team has all the information on time to deliver a working product at the end of the sprint.
3. Acceptance of working packages – delivered working packages are presented to the Client for acceptance at the end of each sprint. The Client shall accept the working package or notify the Consultant of any defects during the following sprint.

Although it is not strictly necessary, the Product Owner may participate in team stand up meetings listening for progress and eventual blockers for an immediate reaction. Product Owner also decides on product releases, as per release plan.

Also, as per the principles of Agile project management methodology, the Client will define the Product Vision Statement and Product Roadmap in order to track progress and to ensure the appropriate product development.

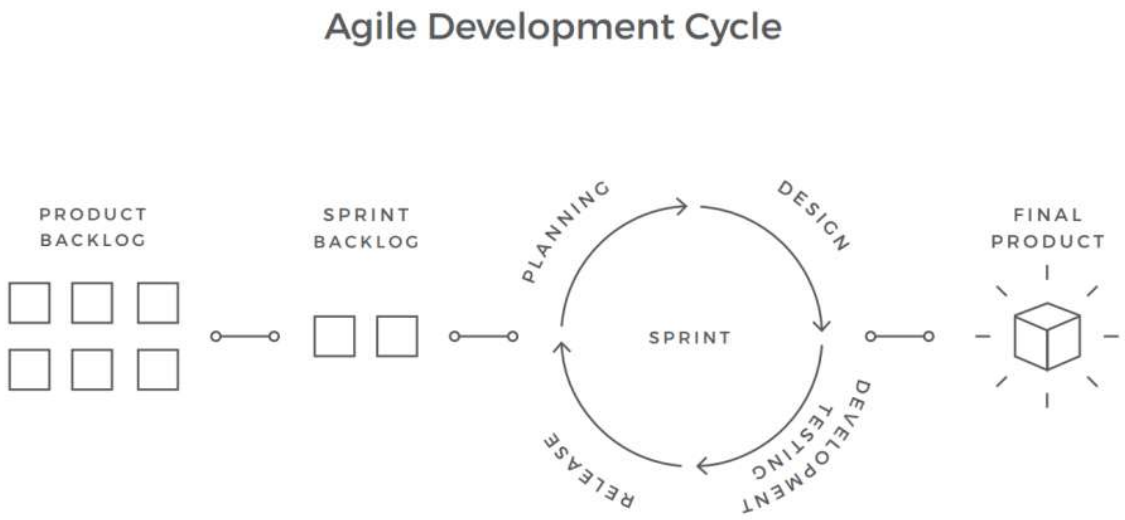


Figure 1. The indicative illustration of the Agile Development Cycle/Process

Warranty

The Consultant shall provide 3 months of warranty for the developed solution. The warranty period starts after final release. During the warranty period the Consultant shall fix any identified defects.

The development and operations must be in compliance with the legal and regulatory documents listed in **Annex 3**.

Required technology stack

To preserve e-Government investments, the solution shall be developed using the latest versions of the following technology stack:

- Programming language is C#.
- ORM is Entity Framework Core.
- Web framework is ASP.NET MVC Core.
- RDBMS is SQL Server.
- Container engine is Docker.
- Container orchestration is Kubernetes.
- Cache server and session store is SQL Server or Redis.

During the development process, the Consultant or the Client may propose use of additional components required for the development and proper functionality of the solution in production. Upon the Client's approval of such components, the costs for them shall be added through amendments to the contract.

4. Expected Deliverables

The following deliverables will be provided by the Consultant during this assignment:

1. **A fully functional information system** with all functionalities developed and deployed according to the requirements defined by the Client during the assignment. The Consultant will deliver compilable and documented source code (including third-party tools and libraries, licenses, where applicable and automation scripts).
2. Technical and End-user documentation developed according to the Client's documentation requirements defined in Annex 2.
3. Training sessions and training materials developed according to the Client's training requirements defined in Annex 2.

Please note that any population with or migration of data is not part of this assignment.

5. Reporting Requirements

The following reports will be provided during the assignment:

a) Sprint Report, including release notes, breakdown and duration of tasks implemented during the sprint, velocity, issues and outstanding problems, proposed actions to be taken;
b) Next Sprint Backlog, including breakdown and estimated duration of tasks proposed to be implemented during the next sprint, resources that the Consultant expects to be provided by the Client and/or actions to be taken by the Client;
c) Training reports, submitted after each training session, including:
- Participants list;
- Training session agenda;
- Training materials (presentations, labs etc.);
- Trainees test results.

6. Timing

The tasks defined under the current contract are estimated to be performed in *12 months – 9 months for development and 3 months of warranty period*. If new functionalities will be identified by the Client based on users' feedback and subject to satisfactory performance, the contract can be extended based on the same fee rates.

7. Institutional arrangements

The Client is responsible for all administrative and procedural aspects, contract and financial management, including acceptance and payment of deliverables/reports expected under the Contract, general project responsibilities and efficient coordination with stakeholders.

A Product Owner will be appointed by the Client and will coordinate and decide on all issues related to the technical elements of the Contract. The Product Owner will issue the administrative notice on the start date of the implementation of the contract and other administrative duties.

The Client will provide the following:

- infrastructure resources for testing and production environments;
- code repository, issue tracking system, CI/CD environment, task management system via the Client's subscription in Azure DevOps. The Consultant shall not include Azure DevOps subscription in its financial proposal;
- Training facilities.

The **Consultant** will ensure that adequate working conditions (workspace/office premises for experts, office equipment, computers, communication facilities, etc.) and services are provided to the Consultant's staff during the lifetime of the project.

The Consultant will be responsible for the day-to-day management of the project team and availability of necessary resources.

The Consultant will organize the Kick-off meeting and initial Backlog discussion at its premises. All Consultant's Key Experts as specified in the section defining the qualification requirements, shall participate in the Kick-off meeting and initial Backlog discussion. The costs associated with the Client's presence at the Kick-off meeting will be covered by the Client and shall not be included in the Consultant's financial proposal.

The Consultant will ensure visits to the Client site to provide training to end users.

In case the deliverables contain defects and/or there are delays for reasons not imputable to the Client that may impact project outcome, the Consultant may be requested to visits to Client's site in order to solve the project issues.

The communication languages will be Romanian or English.

The Consultant shall work under the supervision of the appointed Product Owner and report to the Client's Chief Digital Officer.

8. Qualification Requirements

Consultant qualifications requirements

The Consultant shall furnish documentary evidence (including information about the completed contracts and contact information of clients from whom the references could be taken or whom the Client may, when necessary, visit to familiarize themselves with the systems put into operation by the Consultant) to demonstrate that it meets the following experience requirements:

1. Have been in operation for at least five (5) years with main part of its business being the development of information systems.
2. Experience in conducting projects similar size and complexity developing web applications proven by at least two (2) contracts with the development phase finalized in the last three (3) years. For ongoing projects, copies of acceptance documents of the entire software solution shall be provided.

3. Experience in software development using agile software development principles (as described in the scope of work and development approach section of the ToR) would be an asset. This shall be demonstrated by presenting the project methodology describing the role of the client.
4. Demonstrated experience using required technology stack would be an asset.

Staff qualifications requirements

The Consultant shall provide a team of the following key experts:

- Key expert 1. *Senior software developer*
- Key expert 2. *Software developer*
- Key expert 3. *Software developer*
- Key expert 4. *Software developer*
- Key expert 5. *Software Tester*
- Key expert 6. *Trainer*

Each key expert must meet at least one the following requirements:

- Proven experience in web UI (user interface) design and development using responsive frameworks, progressive web apps
- Proven experience in database design, development, and optimization
- Experience in systems' integration, API design and development using SOAP/REST
- Experience with unit testing
- Experience in DevOps practices
- Experience in system analysis.

Per total the entire team of the proposed key experts must meet all the above requirements. Offers which will not demonstrate that the team covers the above requirements may be subject of disqualification.

For proposed key experts the CVs need to be submitted, demonstrating the minimum qualifications requirements, as detailed below:

Key Expert 1. *Senior software developer*:
The Senior software developer shall oversee that all reporting obligations are fulfilled in a timely manner to a high-quality standard.

- University degree in Computer Science or another relevant domain
- At least 7 years of experience in software development
- Participated in at least 2 software development projects in the last 3 years using agile approach
- At least 3 years of experience in software development using C#, Entity Framework, ASP.NET MVC, SQL Server and a dependency injection framework
- Certifications in any technology from the required technology stack is an asset
- Ability to communicate in Romanian or English

Key Expert 2-4. *Software developer(s)*:

- University degree in Computer Science or another relevant domain

- At least 5 years' experience in software development
- Participated in at least 2 software development projects in the last 3 years using agile approach
- At least 3 years of experience in software development using C#, Entity Framework, ASP.NET MVC, SQL Server and a dependency injection framework
- Certifications in any technology from the required technology stack is an asset
- Ability to communicate in Romanian or English

Key Expert 5. *Software Tester*:

- University degree in Computer Science or another relevant domain
- At least 3 years' experience in software testing in projects of similar complexity
- Proven experience in software testing analysis and design
- Proven experience in automated testing
- Proven experience in performance (load and stress) testing
- Proven experience in security testing
- Certification in testing or any technology from the required technology stack is an asset
- Ability to communicate in Romanian or English

Key Expert 6. *Trainer*:

- University degree in Computer Science or another relevant domain
- Proven experience in conducting training sessions for end-users and IT specialists in at least 2 similar projects
- Proven experience in writing technical and end-user documentation
- Experience in on-line training development using Moodle e-learning system is an asset
- Ability to communicate in Romanian
- Knowledge of English is an asset

Annexes

Annex 1. Business requirements

1.2 Actors

The following actors will be involved in MDelivery and will act in different capacities:

MDelivery Administrator
An user of MDelivery performing technical administration tasks, acting on behalf of E-Governance Agency.
Shipper
A public service provider and owner of an information system who will integrate with or use MDelivery to deliver physical parcels to individuals and legal entities and is responsible for packing, labeling and preparing all the goods to be delivered, as well as handling all documents and paperwork needed.
Carrier
A entity that transports physical goods (here-and-after reffered to parcels or simply goods) and is responsible for pick-up, sort, store, ship, drop-off and tracking of parcels with various delivery options (same-day, next business day or other time, express, priority, international).
Receiver
The final identity receiving goods from public service providers through MDelivery, explicitly specified by the Shipper or identified by Identity Resolver as a result of its relationships with other identities.

1.2 Main information workflow

The main workflow of MDelivery is as follows:

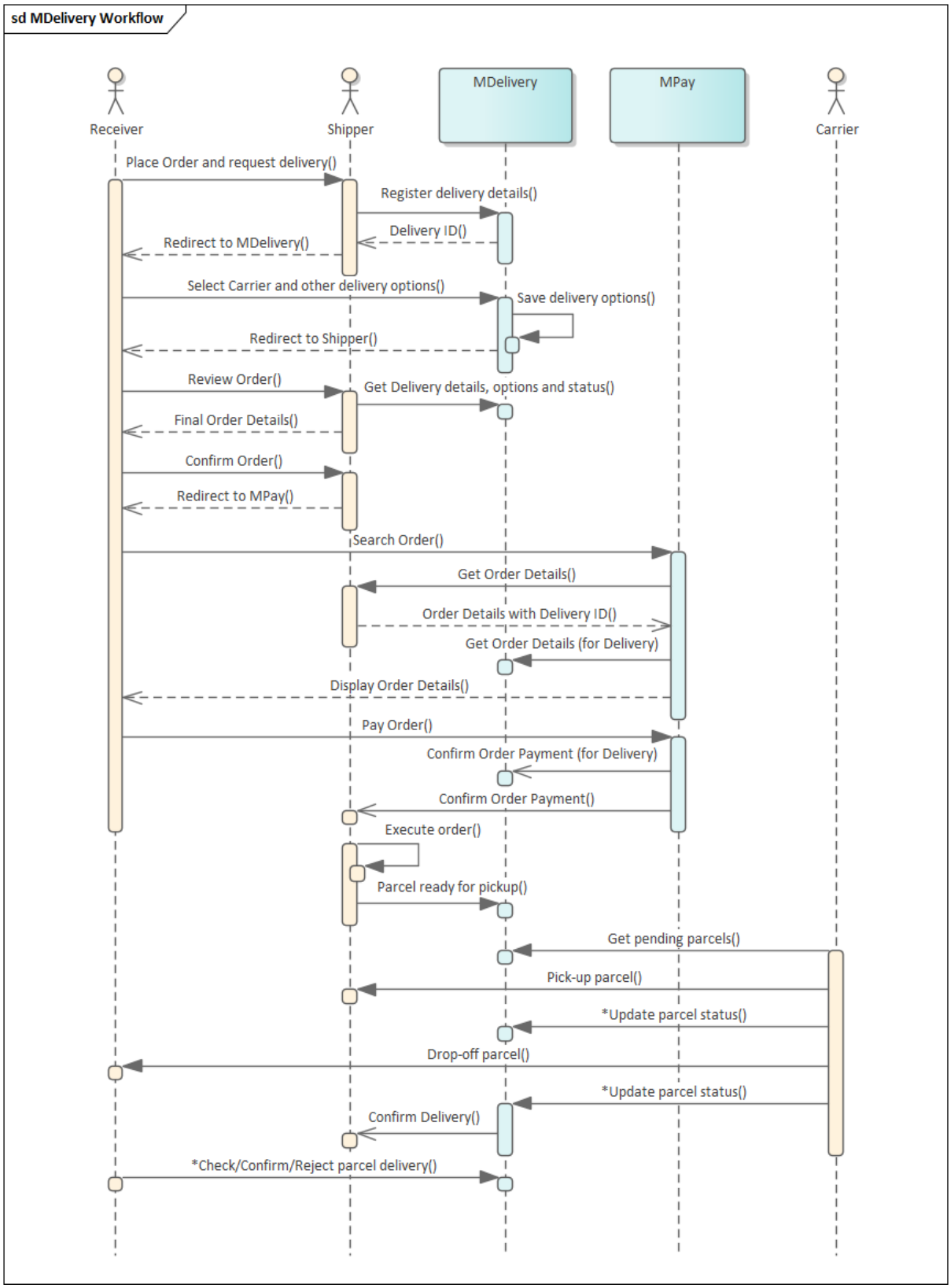
1. The Receiver starts with placing an order at a public service provider and requesting the delivery of the goods.
2. The delivery details (such as sizes, weights, and quantities of goods, drop-off address, etc.) are submitted to MDelivery API by the Shipper information system (IS).
3. MDelivery returns a Delivery ID which is saved by the Shipper for further usage.
4. Shipper redirects the Receiver to MDelivery providing the Delivery ID.
5. Receiver selects the Carrier and provides other delivery options.
6. MDelivery saves the delivery options for further processing and redirects the Receiver to Shipper.
7. Receiver reviews the final order, which includes some details about the delivery (including Carrier, cost, estimated shipment date, etc.) retrieved from MDelivery API.
8. Receiver confirms the order to the Shipper, which redirects to MPay for payment.
9. MPay retrieves main order details from the Shipper, including details related to additional order services, which include Delivery ID.
10. MPay retrieves order details for additional services from their providers, in this case MDelivery.

11. Having payment details confirmed, MPay confirms order payments first to additional order services (including MDelivery) and for the main order to the Shipper.
12. After executing the order and preparing the parcel, Shipper notifies MDelivery about parcel ready for pickup.
13. Depending on their schedule, Carrier retrieves pending parcels from MDelivery API.
14. If there are any parcels pending, Carrier picks up the parcels from the Shipper and updates parcel status through MDelivery API. The status might be updated multiple times for tracking delivery progress.
15. After dropping-off a parcel, Carrier updates the status of the parcel through MDelivery API, which in turn confirms the delivery to the Shipper (if a callback address is provided).
16. Receiver can check the status of the delivery any time he wants. After the drop-off, Receiver can optionally confirm or reject the delivery. If the Receiver does not confirm the delivery, MDelivery will implicitly confirm it after some time (e.g. 14 days).

In case of Shipper not having an IS, all interaction might be accomplished by a Shipper Operator after authenticating into MDelivery UI. This includes registering delivery details, configuring delivery options, paying for the actual delivery order, and confirming when parcel is ready for pickup.

In case of Carrier not having an IS, all interaction might be accomplished by a Carrier Operator after authenticating into MDelivery UI. This includes checking for pending parcels and updating parcel delivery status.

The following diagram highlights the processes and functions that MDelivery either supports or triggers in external components.

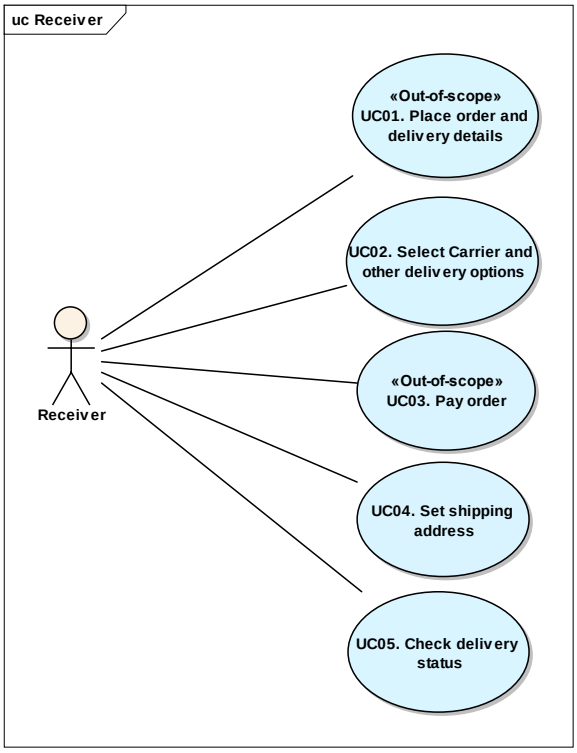

sd MDelivery Workflow
Receiver
Shipper
MDelivery
MPay
Carrier
Place Order and request delivery()
Register delivery details()
Delivery ID()
Redirect to MDelivery()
Select Carrier and other delivery options()
Save delivery options()
Redirect to Shipper()
Review Order()
Get Delivery details, options and status()
Final Order Details()
Confirm Order()
Redirect to MPay()
Search Order()
Get Order Details()
Order Details with Delivery ID()
Get Order Details (for Delivery)
Display Order Details()
Pay Order()
Confirm Order Payment (for Delivery)
Confirm Order Payment()
Execute order()
Parcel ready for pickup()
Get pending parcels()
Pick-up parcel()
*Update parcel status()
Drop-off parcel()
*Update parcel status()
Confirm Delivery()
*Check/Confirm/Reject parcel delivery()

MDelivery workflow may include order cancellation, delivery rejection, delivery address change, which will be analyzed during the sprints and will be included in MDelivery product back-log.

1.3 Use cases

1.3.1 RECEIVER USE CASES

This section describes Receiver use cases.



UC01: Place order and request delivery (out-of-scope)
Receiver can place and order on Shipper IS and request delivery. When Receiver places the order and delivery request, the Shipper collects the delivery details relevant at this stage (such as sizes, weights, and quantities of goods, drop-off addresses etc.).

UC02: Select Carrier and other delivery options
Receiver can select Carrier and delivery options such as delivery options (same-day, next business day or other time, express, priority, international), by being redirected from Shipper IS to MDelivery UI. The selection of Carrier and options will influence the final price of the delivery, as each Carrier will have its own options and associated prices.

UC03: Pay order (out-of-scope)
Receiver can pay the placed order including delivery price by being redirected to MPay service. In MPay, the Receiver will select payment method and pay the order and the delivery price accordingly.

UC04: Set shipping address

Receiver can set shipping address in its profile by accessing MDelivery UI and authenticating through MPass. Once the shipping address is set in MDelivery, the Shipper may no longer send the drop-off address when registering delivery details and send instead the Receiver identity.

UC05: Check delivery status

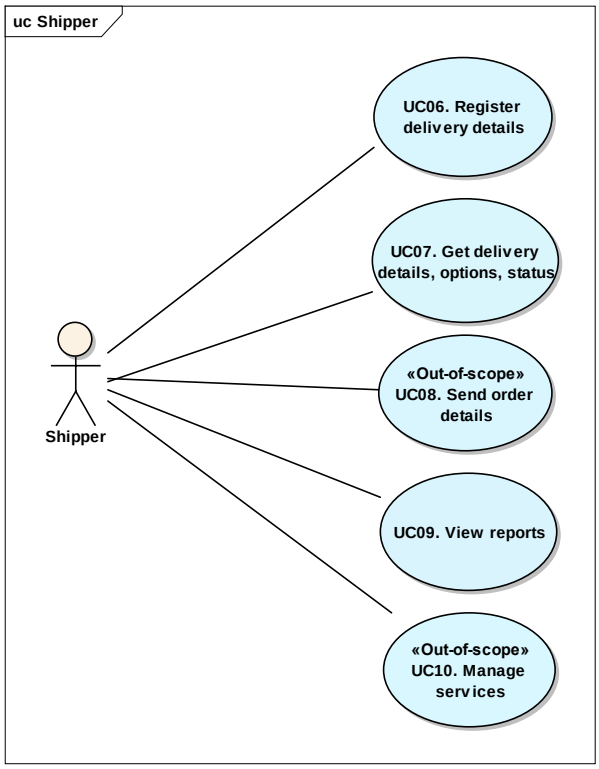
Receiver can check delivery status in MDelivery UI. MDelivery will respond with delivery status. The following delivery status may apply:

- Pending – the parcel has been created for informational purposes and inventory has been reserved. However, it has not begun the delivery process.
- Waiting – the parcel is waiting for packing and labelling.
- Ready for pick-up – the parcel is ready to begin the pick-up process.
- Transmitted – the parcel has been transmitted to Carrier for shipment.
- In transit - the parcel is in transit to arrive at its destination.
- Delivered – the parcel has been delivered to Receiver.
- Declined — Carrier has marked the parcel as declined by Receiver.
- Returned — Carrier has returned the parcel to Shipper because he was unable to deliver it.
- Refunded — payment for delivery has been refunded to Receiver because he canceled it.
- Canceled – delivery order has been canceled. Delivery order may be canceled only before the pick-up is initiated by the Carrier.
- Failed – delivery order request failed, and failure code is included in the status response.

MDelivery will also notify the Receiver about the delivery status using MNotify capabilities.

1.3.2 SHIPPER USE CASES

This section describes Shipper use cases.



UC06: Register delivery details
The Shipper can register delivery details to MDelivery by submitting a request to MDelivery by calling its API or by filling out manually/importing delivery details using MDelivery UI (user interface).
The delivery request contains information about the parcel (such as sizes, weights, and quantities of goods), Receiver identity (in case the Receiver has set the shipping address in its profile in MDelivery), pick-up address (if the address is not set in MPass).
MDelivery returns a Delivery ID which is saved by the Shipper for further usage.

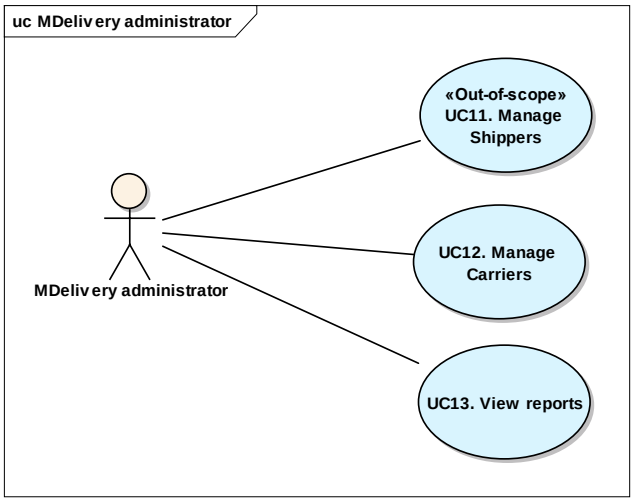
UC07: Get delivery details, options, status
After the Receiver selects Carrier and provides other delivery options, such as delivery timeframe in MDelivery, Shipper calls MDelivery API to request delivery details, options, status in order to display them to Receiver so that he can confirm the final order.

UC08: Send order details (out-of-scope)
After the Receiver confirms the final order, Shipper redirects the Receiver to MPay. MPay calls Shipper IS and retrieves order details and delivery ID. MPay calls MDelivery API by indicating delivery ID and retrieves delivery price.

UC09: View reports
The Shipper can view and download MDelivery statistics related to his/her usage in MDelivery UI.

UC10: Manage services (out-of-scope)
The Shipper can manage its services (to be integrated with MDelivery API) in MPass. MPass services registration includes service name, service owner, digital certificates, other specific settings (e.g. set pick-up address once per service etc.).

1.3.3 MDELIVERY ADMINISTRATOR USE CASES
This section describes MDelivery administrator Use Cases.



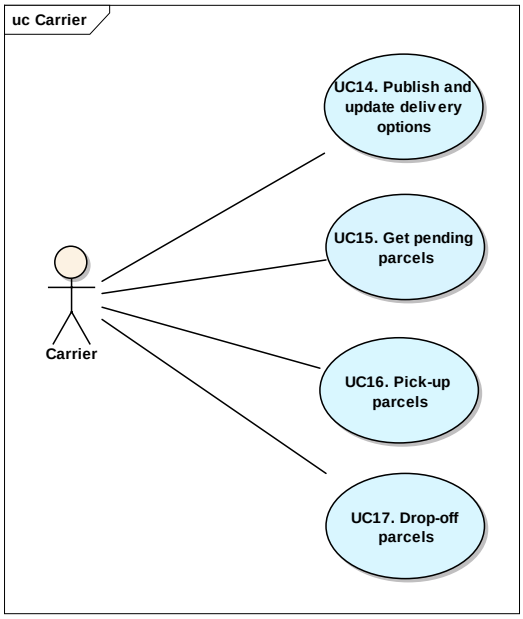
UC11: Manage Shippers (out of scope)
MDelivery administrator can manage Shipper in MPass including its IS settings related to MDelivery (as described in UC10).

UC12: Manage Carriers
MDelivery administrator can manage Carriers configurations.

UC13: View reports
MDelivery administrator can view and download MDelivery statistics and billing reports.

1.3.4 CARRIER USE CASES

This section contains Carrier Use Cases.



UC14: Publish and update delivery options
Carrier can publish and update delivery options (delivery timeframe, express, priority, national, international etc. as well as the prices thereto) to MDelivery by calling its API.

UC15: Get pending parcels
Depending on its schedule, Carrier can retrieve pending parcels form MDelivery API so that he can send its courier to pick-up the parcel.

UC16: Pick-up parcels
If there are any parcels pending, Carrier picks up the parcels from the Shipper and updates parcel status through MDelivery API. The status might be updated multiple times for tracking delivery progress (pick-up, store, sorting, in transit etc.).

UC17: Drop-off parcels

After dropping-off a parcel, Carrier updates the status of the parcel through MDelivery API, which in turn confirms the delivery to the Shipper (if a callback address is provided).

1.3.5 GENERAL BUSINESS REQUIREMENTS

BR01: User authentication
MDelivery users will be authenticated and authorized through MPass.

BR02: Shipper IS authentication
Shipper IS will be managed in MPass and their profiles will be retrieved by MDelivery when needed.

BR03: Log business events
MDelivery shall log details about all important business events (e.g. sending delivery details, canceling deliveries, rejected deliveries, delivered orders etc.) in MLog.

BR04: Carrier implementation
The Consultant shall implement integration with at least two Carriers (Posta Moldova and one private Carrier) as part of this assignment.

BR05: Shipper integration
The Consultant shall implement integration with at least two Shippers as part of this assignment.

Annex 2. Technical requirements

1. Documentation requirements

User Documentation	The Consultant will prepare and deliver the following documentation for end-users: Interactive guidance included in user interface of MDelivery adjusted to user role (Shipper, Carrier, MDelivery administrator etc.) Downloadable user manuals in PDF format for Shipper, Carrier, MDelivery administrator etc. All end-user documentation will be provided in Romanian.
How-To video tutorials	The Consultant will prepare How-To video tutorials for MDelivery main functions (e.g. delivery order, carrier, shipper, configuration etc.). The tutorials will be provided in Romanian.
Technical documentation	The Consultant will prepare and deliver the following technical documentation: • System architecture documentation (including description of models in UML language, which will include a sufficient level of details of the system architecture) • Test strategy • Compilable and documented source code for applications, components and unit tests developed within the project • System installation and configuration manual (including code

	compilation, container image build scripts, system installation, hardware and software requirements, platform description and configuration, backup and disaster recovery procedures) All technical documentation will be provided in English.
API documentation	The Consultant will prepare and deliver: • API integration guide • Integration samples in .NET, Java, PHP, Python • Human and machine-readable description in a standard description language (e.g. WSDL or Swagger). All API documentation will be provided in English.

2. Training requirements

Training sessions	The Consultant will provide on-line training sessions using developed e-learning modules based on Moodle LMS for the following target groups: • MDelivery Administrators from eGA and STISC. • Shippers. • Carriers.
Training materials	Training documentation – curricula, training courses (manuals, video tutorials, quizzes, etc.) for administrators, Shippers, Carriers and end-users (individuals and businesses) developed in e-learning platform based on Moodle LMS. All training content/materials will be provided in Romanian.

3. Rights requirements

Perpetual software license	The Consultant grants to the Client the rights to run and use entire solution with all included software components with no constraints on time, location and offered functionality.
Redistribution rights	The Consultant shall grant to the Client the right to re-distribute the solution. While the Client does not intend to re-distribute at a massive scale it still envisions the need to transfer the software solution to another state agency due for example to potential reorganization. Also, the Client might get the opportunity to re-deploy the entire e-Government platform elsewhere.
Full data rights	The Client keeps full rights on data created by the means of this solution.
Open data format	The solution preserves the data in an open format or includes mechanisms to extract data from the system in an open format thus enabling the capability to transfer/migrate the data into another system.

4. Architecture requirements

Open standards	The solution architecture shall be based on relevant open standards. The solution architecture shall not use proprietary standards.
Service Oriented Architecture	The solution shall be based on a Service Oriented Architecture.
Hosting environment	The solution shall not include any hardware components and upon finalization will be deployed on governmental cloud environment (MCloud).
Running environment	System shall run on Docker container engine and shall not depend on specific host OS instance. Building container images shall be automated. (refer to the following link for details: https://docs.docker.com/develop) Running in a container-based environment, the application must be elastic, including when adding/removing application container instances (above minimum required instances for HA), changing of configurations and system parameters has no impact on any work in progress, such as any active sessions, requests, etc.
Multiple sites	The solution architecture shall ensure high availability including during new versions deployment and the possibility to run simultaneously on multiple sites
Browser compatibility requirements	The system shall be compatible with latest two major versions (to be considered at the time of system acceptance) of following web browsers: Chrome, Safari, FireFox and Edge.
API for integration with governmental platform services and third-party systems	MDelivery shall implement API to be consumed by governmental platform services (e.g. MCabinet) and by third party systems. The full list of logically applicable APIs and their format will be detailed during analysis and design stages.
Detailed data model	System's detailed data model shall be described fully in a machine-readable data scheme for example using a DDL language for relational databases. The Consultant shall coordinate the detailed data model schema format with the Client in advance.

5. System Integration requirements

Governmental platform services integration	MLog shall be used to journal business critical events. The events that are business critical will be defined at analysis and design stages and must be configurable. MPass shall be used to retrieve contact data and notification preferences of the

	Receivers. MDelivery will use MNotify for notifying purposes.
Open data integration	MDelivery shall publish agreed sets of data in a machine-readable format to Open Data portal located at date.gov.md using its API.

6. System Performance requirements

Asynchronous processing	System shall use asynchronous processing whenever possible to perform any input-output.
Concurrent users	The system standard load and performance shall be guaranteed for 100 concurrent human users.
Concurrent system requests	The system shall be designed to respond (via API requests) to at least 1000 concurrent external system requests.
Response time	Response time for system functions shall be under 3 (three) second. The Consultant shall list the exceptions, if any, and discuss/agree them with the Client at analysis and design stages.
Daily transactions	The system shall be designed to process at least 10000 transactions per day.
Key performance Indicators	The system shall meter and expose its key performance indicators. The Consultant shall propose the list of indicators and discuss/agree them with the Client.

7. User Interface requirements

Multilanguage User Interface	The system shall support multilanguage user interface. This support includes data type specific formats (such as date, time, time spans, currencies, etc.). The system front-end interface will be delivered with at least Romanian, Russian and English interfaces. The system back-end shall be delivered at least in Romanian and English. The default language for User interface shall be the Romanian.
User Interface accessibility	User interface shall conform at least to Level A of Web Content Accessibility Guidelines 2.0. https://www.w3.org/TR/WCAG20/
Responsive/Adaptive design	The system user interface shall automatically adapt to various display resolutions. Minimal display width is 480px. The system's UI shall be implemented using progressive web application (PWA) technologies and shall be functional on mobile devices.
Contextual help	User Interface elements shall include Tips and Hints for user interface elements.
Client support	All pages shall include client support contacts.
Bookmarks	All major MDelivery pages shall be bookmarkable and the User shall be able to access bookmarked pages later. The bookmarkable pages will be defined at analyzing stage.
Friendly URLs	MDelivery shall use friendly URLs for accessing its pages.

8. System maintenance requirements

System logs	The system shall log its various actions and events in a structured manner. Logging shall be configurable and based on extensible logging framework (such as log4net, nlog, etc.). Logging framework shall minimally support JSON format and the following targets: console, rolling files, UDP and HTTP POST.
Log levels and event log records	The system shall differentiate events and actions it logs into at least following levels: Critical, Error, Warning, Info, Debug Critical and Error level events shall be logged only for non-recoverable error that requires human intervention. Event log records will include at least: the type of the event timestamp when the event took place event level system component that produced the event user/user agent, IP that triggered the event information object identifier affected textual details about the produced event
Graceful shutdown	The system shall implement graceful shutdown, i.e. shutting down an application container instance at any time shall not impact any work in progress, such as any active sessions, requests, event logs, etc.
Source code	The Consultant shall supply all the source code for system components that are not available as COTS from third parties. The source code shall use package managers for dependencies to 3rd party libraries. All prerequisite software must be part of container image definition and based on public container repository.
System deployment	The Consultant shall supply the deployment procedure and supporting tools for this. Deployment procedure shall cover all the prerequisites before proceeding to system installation. The deployment shall be automated and include database structure initialization and seeding.
System upgrades	System upgrades shall be automated, including database upgrade/downgrade scripts or code. To enable rolling upgrades in production environment, the recommended practice is to perform database breaking changes in incremental changes.

9. Security requirements

Secure architecture	The system shall be secure by design and comply with the relevant requirements specified in GD 201 from 28.03.2017 (http://lex.justice.md/md/369772/). The Consultant shall supply documentation describing this design and supporting evidences that such a design is secure. Note that the Consultant will coordinate with the Client the format of the documentation, supporting evidence and list of requirements to comply with.
Least privilege principle	The system's components shall rely on the least privilege principle and run under such a limited privilege account under the OS rights model.

enforcement	The documentation shall highlight each of the system's components required privilege level and considerations that force use of that level or access.
Secrets and addresses	Secrets (passwords, private keys and certificates, connection strings) and addresses of external services shall be clearly delineated in configuration documentation and easily modifiable via automated scripts.
Secure communication channels	All system's communication with external systems or users takes place over encrypted communication channels.
No Username/Password authentication	The system shall rely on authentication via MPass. Other forms of user authentication shall not be used.
Minimize personal information storage	The system shall minimize the amount of personally identifiable information stored. For example, there is no need to store a user's First and Second names since this will be provided after authentication by MPass. The system shall comply with the relevant requirements related to personal data processing. Note that the Consultant shall coordinate with the Client the list of requirements to comply with.
Secure against OWASP Top 10 vulnerabilities	The system shall include security controls for all its components for at least OWASP Top 10 vulnerabilities. Refer https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project
Health-check API	The system shall expose readiness and health-check API via a HTTP GET requests. The health-check shall check the health of as many system components as possible. In case of health check error, a human-readable error message shall be returned.
Users' roles management	The users and their roles will be managed in MPass. The system shall retrieve the users' roles from MPass.
Session expiration	The system shall include a session expiration mechanism when after a specific period of inactivity, the user is required to authenticate again. The period of inactivity shall be configurable and by default it is 15 minutes.
Authorized access to personal content	Users are granted access to content designated as belonging to them. Content belongs to a user if it has been assigned/addressed to their personal IDNP.
Input validation	All input data shall be validated on client and server side.
User content	User content can be captured in text format only. The system shall forbid entry of special characters used for formatting and markup of special Web content. Otherwise all UNICODE characters shall be possible to enter/view by system's components.
Unauthorized access attempts	Unauthorized access attempts When the system registers unauthorized access attempts it shall: log such attempts with at least ERROR level provide users with a warning message that access is not authorized and that abuse

	will be investigated
Data integrity	The Consultant will ensure data integrity by providing appropriate solution for prevention of unauthorized internal activities (for ex. deletion or alteration of notifications directly from database).

10. Support and Warranty requirements

Support	During the warranty period the Consultant shall provide necessary technical assistance to the Client;
Warranty	During the warranty period the Consultant shall: fix all defects reported by the Client; solve all incidents reported by the Client according to the agreed SLAs; Note: The response and resolution time shall not exceed 60 minutes for non-critical errors and 15 minutes in case of critical errors. The incidents shall be solved within 2 working days for non-critical errors and within 4 working hours for critical errors starting from escalation time. Hourly progress report will be provided for critical errors.

Annex 3. Relevant legal acts and regulations

1. Law nr.91/2014 on electronic signature and electronic document - http://lex.justice.md/index.php?action=view&view=doc&lang=1&id=353612.
2. Law nr.71/2007 on registries - http://lex.justice.md/index.php?action=view&view=doc&lang=1&id=325732.
3. Law nr.1069/2000 on informatics - http://lex.justice.md/index.php?action=view&view=doc&lang=1&id=312902.
4. Law nr.467/2003 on informatics and state informational resources - http://lex.justice.md/index.php?action=view&view=doc&lang=1&id=313189.
5. Law nr.982/2000 on access to information - http://lex.justice.md/index.php?action=view&view=doc&lang=1&id=311759.
6. Law nr.133/2011 on personal data protection - http://lex.justice.md/index.php?action=view&view=doc&lang=1&id=340495.
7. Law nr.142/2018 on data exchange and interoperability - http://lex.justice.md/index.php?action=view&view=doc&lang=1&id=376762.
8. Government Decision nr.710/2011 on approving strategic Programme of technological modernization of government (e-Transformation) - http://lex.justice.md/index.php?action=view&view=doc&lang=1&id=340301;
9. Government Decision nr.1140/2017 on approving the Regulation of the activity of the certification service providers in the field application of the electronic signature - http://lex.justice.md/md/373494/.
10. Government Decision nr.1141/2017 on approving the Regulation on modality of application of the electronic signature on electronic documents by functionaries of legal persons governed by public law in the electronic document circulation - http://lex.justice.md/md/373495/.
11. Government Decision nr.128/2014 on Government single technological platform (MCloud) - http://lex.justice.md/index.php?action=view&view=doc&lang=1&id=351760.
12. Government Decision nr.1090/2013 on the governmental electronic service of authentication and access control (MPass) - http://lex.justice.md/index.php?action=view&view=doc&lang=1&id=351035.
13. Government Decision nr.405/2014 on the governmental electronic integrated service for digital signature (MSign) - http://lex.justice.md/index.php?action=view&view=doc&lang=1&id=353239.
14. Government Decision nr.708/2014 on the governmental electronic journaling service (MLog) - http://lex.justice.md/index.php?action=view&view=doc&lang=1&id=354589.
15. Government Decision nr.916/2007 on the concept of a Government Portal - http://lex.justice.md/index.php?action=view&view=doc&lang=1&id=324962.
16. Government Decision nr.330/28.05.2012 on development and administration of a single Service providers portal - http://lex.justice.md/index.php?action=view&view=doc&lang=1&id=343406.
17. Government Decision nr.701/2014 approving the Methodology of government open data publication - http://lex.justice.md/index.php?action=view&view=doc&lang=1&id=354534.