Building Block Definition
Registration

Developed by: Frank Grozel, Ingmar Vali, Tambet Artma, Saurav Bhattarai, Dr Ramkumar, Rauno Kulla
In cooperation with GIZ, ITU, DIAL, and the Government of Estonia

Table of Contents

1 Version History

Version	Authors	Comment
1.0.0	Frank Grozel, Ingmar Vali, Tambet Artma, Saurav Bhattarai, Dr Ramkumar, Rauno Kulla. ingmar.vali@gmail.com	Initial revision
1.1.0	Frank Grozel, Ingmar Vali, Tambet Artma, Saurav Bhattarai, Dr Ramkumar, Rauno Kulla. Reviewers: Neil Roy, Aare Lapõnin, Sasa Kovacevic, Fergal Marrinan, Surendrasingh Sucharia.	Applied feedback from technical review.

2 Description

“Registration” is a process through which an applicant gets **information recorded in a registry** and receives a **credential** as a proof of registration, in exchange for providing **information,** with or without **money**.

> The information provided by the applicant consists of **data** and/or of **credentials** issued by public or private entities. **Money** is provided to pay for one or more registration fees/costs.
>
> A registration involves at least **two parties**: an **applicant** who wants to register (something or somebody) and an authorized representative of a **registrar** in charge of registering the data and issuing the credential.
>
> A registration may involve more than two parties: one or more **third parties** can be requested to **assert/confirm the information** provided by the applicant (a notary, a family member, a witness, another public entity, or a non-human entity such as a database); or to receive the **payment** made by the applicant (a bank, a cashier, an online payment service). The **registry** or registries where the information is written can also be considered as a third party.

In some cases, **multiple registration processes** can take place **simultaneously**, i.e. the same applicant gets information registered in various registries and receives various credentials, while providing information and money only once. This is known as “**single window**”, “simultaneous registrations” or “integrated registrations”.

In practice, from the **applicant’s point of view**, a registration process can entail the following operations:

- Answer questions to determine:
  - if the applicant is eligible to register

  - the data, credentials/documents and fees required, according to the applicant's case
- Provide data in a form, upload copies of credentials/documents
- Pay fees at a physical point of payment (bank, cashier, etc.) or through an online service
- Confirm his/her will to register
- Query the status of the registration process
- Receive credentials electronically or collect them at a physical point of collection

From the **point of view of the entity in charge (Registrar)** of the registration, the process can include the following operations:

- Ask questions to the applicant, on one or more screens, to determine, according to the answers:
  - if the applicant is entitled to register
  - the data, documents and fees required, according to the applicant's case
- Display, on one or various screens, the necessary fields for the applicant to enter the required data and documents (including a receipt of payment, in case the system does not propose an online payment option or if the user for some reason wants to make the payment physically or directly / in the traditional way).
- Control that the information entered is complete and correct (and the applicant is entitled to register)
- Validate the information submitted or, in case the information entered is incomplete or incorrect, send the file back to the applicant for correction
- When possible, call an external application allowing the applicant to pay the fees online
- Record the applicant's information in a registry
- Issue a credential as a proof of registration
- Ensure that there is tracking of all changes to records

3 Terminology

Several elements of the glossary are based on the terminology defined by the World Wide Web Consortium (W3C) recommendations on the "Verifiable Credentials Data Model 1.0" (Nov. 2019) and Open ID Connect Core 1.0.

Registration

Process through which an entity gets claims recorded in a registry and a credential proving the registration, in exchange of providing some requirements.

Entity

A thing with distinct and independent existence, such as a person, organization, or device.

Claim

An attribute asserted by an entity, about itself or another entity.

Attribute

A property (data, information) relating to an entity

Subject

An entity obliged or entitled to a registration, or about which a claim is made

Asserter

An entity that asserts a claim.

Registry

A paper-based or electronic database (centralized or decentralized, i.e. blockchain) where claims are stored and can be consulted.

Registrar

An entity that is authorized to register, in a registry, claims submitted by an applicant and to issue a credential proving the registration.

Applicant

Entity that requests the registration of claims in a registry.

Operator

A registrar or a staff of a registrar who is processing the request of an applicant.

Credential

A paper or electronic document created by an issuer and displaying one or more claims about one or more entities

Issuer

Entity which creates a credential for one or more entities

Scope

A set of claims and/or credentials required for a registration

Service

Name given to a registration, or to a combination of registrations which can be undertaken simultaneously, by the entity(ies) in charge of the registration process.

Regulation

Normative text (law, decree, deed, decision, etc.) issued by a public entity

Rule

Any regulation, or part of a regulation, which creates for specific subjects an obligation or a right to register, and defines, for each category of subjects, the results and requirements of the registration.

Requirement

Information (i.e. claims and credentials) and fees which must be provided in a registration process. Requirements may vary according to each subject.

Result

The result of a registration is usually a credential (sometimes called: certificate, license, permit, card, etc.) proving the registration, in addition to the recording of information in a registry,

Determinant

A determinant is an attribute, defined in the rule, used as a filter to determine/trigger if (1) an entity is subject to a registration and/or (2) what requirements this entity must provide to register. Examples:

> *Employers* must register to the Social Security registry (attribute = employer; what is triggered = subject to register at the Social Security registry)
>
> *Foreign companies* must obtain an investment license before investing (attribute = foreign company; what is triggered = subject to register for an investment license)
>
> *Married* traders must provide a copy of their wedding certificate when they register at the business registry (attribute = married; what is triggered = wedding certificate is one of the requirements to register at the business registry)

4 Key Digital Functionalities

Key Digital Functionalities describe the core (required) functions that this building block must be able to perform. These functionalities are described as business processes as opposed to technical specifications or API definitions.

The Registration BB is a software allowing to develop online registration services, without coding (= no-code development platform). It is composed of two main elements:

- A no-code development platform
- The online registration services developed with the platform (there is no limit to the number of services that can be developed)

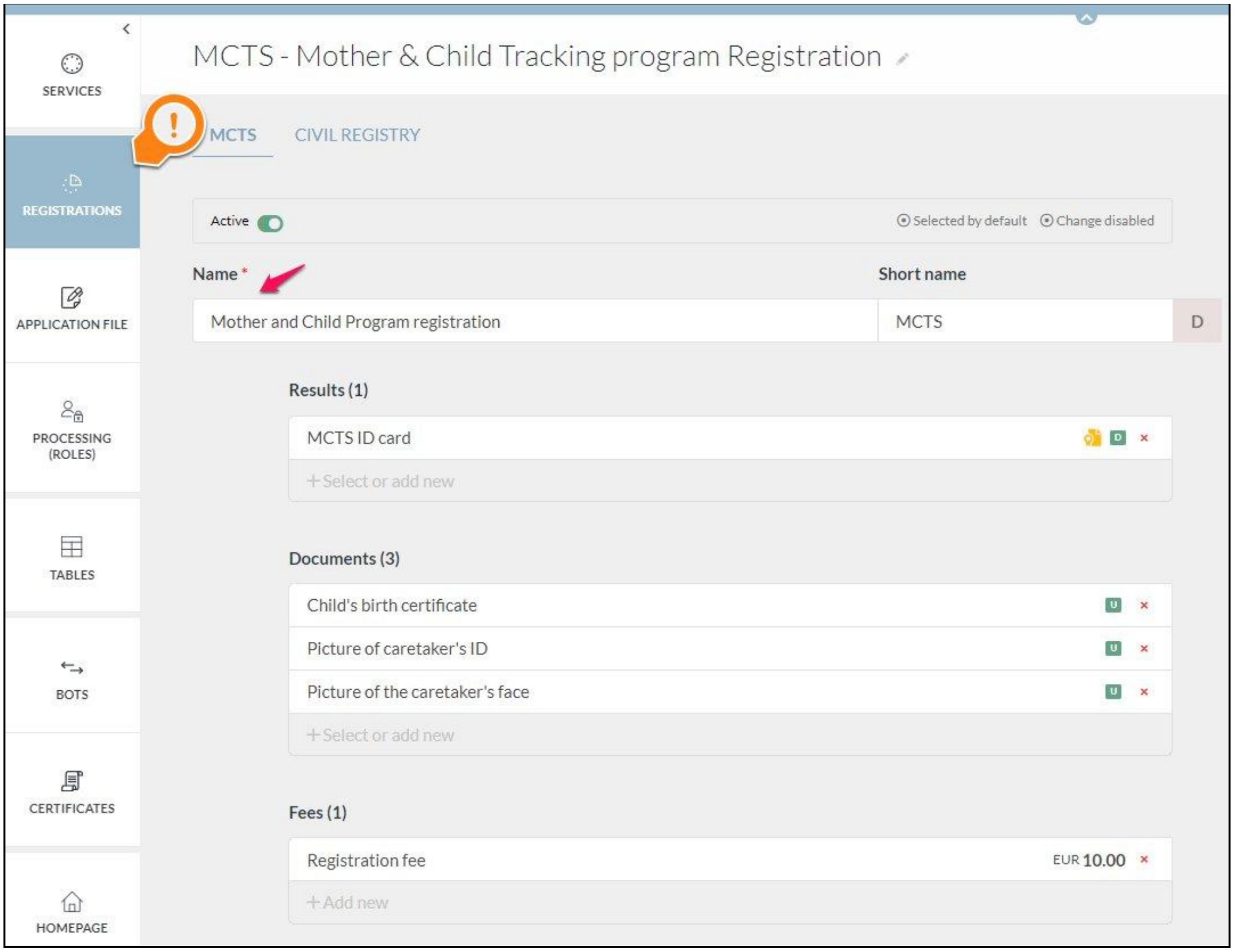
The no-code development platform is used by authorized personnel, called “analysts”, entrusted by the entities in charge of the registrations to develop the corresponding online services.

The online registration services developed with the no code platform are used by the applicants who want to be registered and by the operators in charge of processing the requests.

4.1 No-Code Development Platform

The no-code development platform acts as a rules engine, a screens (user interface) and flow builder and a control configurator:

- **Rules engine**: a tool transforming legal rules relating to a registration (i.e subjects, results, requirements and determinants), defined by a human analyst, into machine-readable statements. See definition and illustration of "rules engine".
- **Screens (user interface) and flow builder**: a tool for a human analyst to create and organize the screens and fields that are necessary for the registration (application file) and processing.
- **Controls configurator**: a tool for a human analyst to define, for each field in the application file and processing screens, what controls will be applied (input format, formulas, actions/checks between fields and with external databases).



MCTS - Mother & Child Tracking program Registration

Active ● ○ Selected by default ○ Change disabled

Name *	Short name
Mother and Child Program registration	MCTS

Results (1)

- MCTS ID card D ×

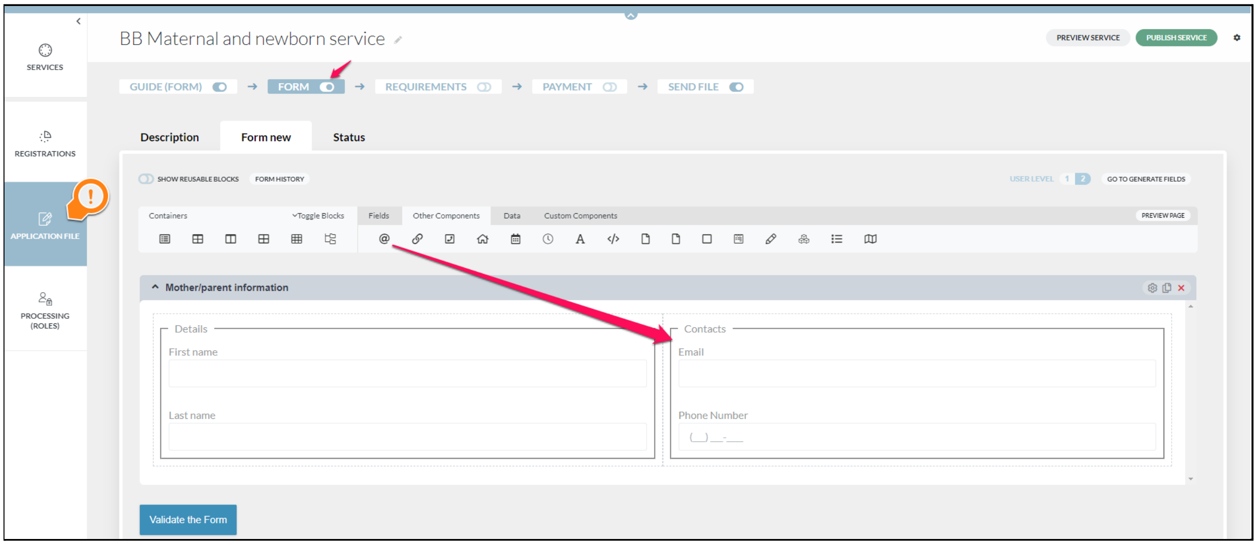
Documents (3)

- Child's birth certificate U ×
- Picture of caretaker's ID U ×
- Picture of the caretaker's face U ×

Fees (1)

- Registration fee EUR 10.00 ×

Illustration 1- Example of configuration of the rules engine (UNCTAD's eRegistrations).



BB Maternal and newborn service

Mother/parent information

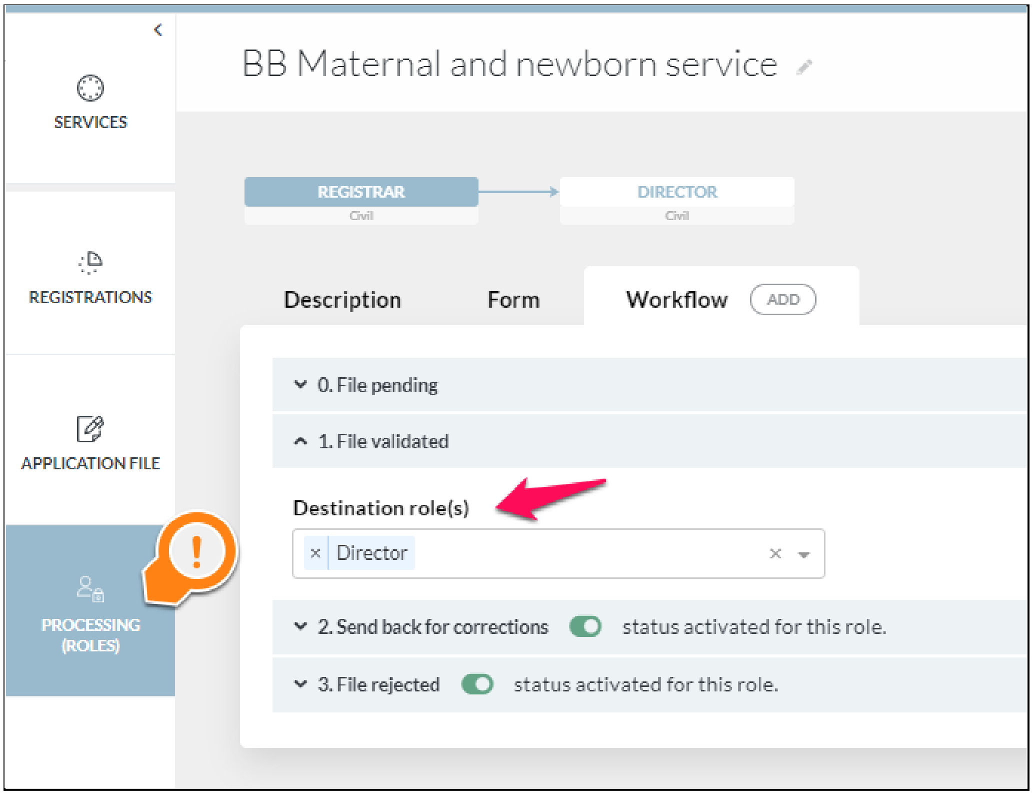
Details

- First name
- Last name

Contacts

- Email
- Phone Number (__) ___-____

Illustration 2 - Example of the user interface builder (UNCTAD's eRegistrations).



BB Maternal and newborn service

0. File pending

1. File validated

Destination role(s): Director

2. Send back for corrections — status activated for this role.

3. File rejected — status activated for this role.

Illustration 3 - Example of flow builder (UNCTAD's eRegistrations).

4.2 The Online Registration Services

The online registration services (e-services) created through the no-code development platform are adapted to any type of registration. They consist in a succession of online screens and actions through which:

- an applicant can provide claims (= fill a form), credentials (=upload documents) and fees (= pay online or upload a payment receipt) and send his/her request to one or more entities in charge of the registration (we call this part "application file")

- an applicant can see/monitor the status of the submitted application file and receive results from the entities in charge of the registration

- Validation: one or more human or automated ("robot" or "BOT role") operators can review the elements provided, approve or reject an application, send claims to a registry and issue a credential (we call this part: "processing")

- For each service, user rights can be defined and statistical reports can be configured ("administration" part)

4.3 Out of Scope Assumptions

- Registration BB does not define which registrant/object should be registered. But allowing applicants to determine if they are or not subject to a registration is part of the registration BB.
- Long term storing of registration data/claims and results is not covered in this BB. See Digital Registries BB for data storage functionalities. However, the connection to the Digital Registries BB is part of the scope.
- Event notification from external endpoints is not covered in this BB. See more in Messaging BB. However, the connection to the Messaging BB is part of the scope.
- Payment solutions are not covered in this BB. See more in Payment BB. However, the connection to the Payment BB is part of the scope.
- Data transfer security solutions are not covered in this BB. See more Information Mediator BB. However, the connection to the Information Mediator must exist as it is the base for the connectivity with other BB-s.
- Consent management is not covered in this BB. See more in the Consent BB. However, the connection to the Consent BB is part of the scope.
- Authentication/login is not managed in this BB. See more Security BB. However, the connection to the Authentication solution must be done in each BB.
- Integration with special hardware (scanners, fingerprint readers, digital signature solutions etc) is not covered in this BB.
- SMS and/or Voice services are not covered during the first iteration.

5 Cross-Cutting Requirements

The cross-cutting requirements described in this section are an extension of the cross-cutting requirements defined in the Architecture specification document (here) and Security requirements (here). This section will highlight important requirements or describe any additional cross-cutting requirements that apply to this building block.

Secure ingress and Egress access mechanisms (Information Mediator and api gateway)

Egress, in the world of networking, implies traffic that exits an entity or a network boundary, while Ingress is traffic that enters the boundary of a network. In the cloud, Ingress means something slightly different. Private networks here refers to resources inside the network boundary of a data center or cloud environment and its IP space is completely under the control of an entity who operates it.

https://aviatrix.com/learn-center/cloud-security/egress-and-ingress/

Inclusion issues - Multiple languages, assisted model, differently abled people, etc.

6 Functional Requirements

The Functional Requirements section lists the technical capabilities that this building block must/could have. These requirements should be sufficient to deliver all functionalities that are listed in the Key Digital Functionalities section. These Functional Requirements do not define specific APIs - they provide specifications and information about any functionalities that must be implemented within the Building Block.

The functionalities described in this chapter apply to any government registration use case. Therefore, they also apply to the following use cases:

1. Postpartum and Infant care- https://solutions.dial.community/use_cases/postpartum_and_infant_care

2. Unconditional social cash transfer https://solutions.dial.community/use_cases/unconditional_social_cash_transf

6.1 No-Code Development Platform

The no-code development platform is composed of:

- Rules engine (6.1.1)
- User interface and flow builder (6.1.2)
- Controls configurator (6.1.3)

It is used by authorized personnel, called "analysts", entrusted by the entities in charge of the registrations to develop the corresponding online services.

Pre-requirements:

- The analyst is authenticated and authorized to use the no-code development platform.
- The analyst has a computer/smartphone with an internet connection.

Post-requirements:

- When the no-code development is finished, the analyst publishes the e-service, which becomes available to users (applicants and operators);
- Applicants can submit applications;
- Operators can process the applications.

6.1.1 Rules Engine Functional Requirements

The **Rules Engine** is a module where an analyst defines:

1. Who are the **subjects** of the registration
2. For each subject, the **result** of the registration, e.g. a credential
3. For each subject, what are the **requirements** of the registration:
    a. Claims required
    b. Credentials required
    c. Payments/fees required

The rules can be:

- Entered in the rule engine by an analyst, on the basis of the regulations
- Provided by external rules providers (e.g. rules DBs at ministries level)

REQ-#	Requirement	Type & UC
RE-1	**Creation of Services** An analyst (user) must be able to create, in the rule engine, one or more services, each service encompassing one or more registrations. A "Service" is a name given to a registration, or to a combination of registrations which can be undertaken simultaneously. To create a service, an analyst will: ● Give a name to the service ● Link one or more registrations to the service	Must have
RE-2	**Publish service to external instance** Each service can be published in the same or in a separate instance, together with the rule engine. Instance must be configured and interoperable with Registration BB service definitions.	May have
RE-3	**Creation of one or more "Registrations"** An analyst can create one or more registrations. For each registration, the analyst defines in clear language, in the rules engine: ● The name of the registration ● The entity in charge of the registration	Must have
RE-4	**Definition of the subjects of a registration** For each registration, an analyst must be able to report/input in the rule engine, in clear language, rules defining who/what are the subjects of the registration.	Must have

REQ-#	Requirement	Type & UC
	To this end, the analyst can select one or various of the following options: 1) The registration is mandatory to all 2) The registration is mandatory to specific subjects 3) The registration is optional to all 4) The registration is optional to specific subjects 1 and 2 can't be selected simultaneously; 3 and 4 can't be selected simultaneously. Specific subjects (in 2 and 4) can be defined through determinants or a combination of determinants. Determinants can be combined by "AND" and "OR" operators. Combinations can be grouped into "groups of determinants". Group of determinants can be combined through "AND" and "OR" operators. Examples: a) Registration is mandatory for attribute "resident" (all residents must register) b) Registration is mandatory for attribute "resident" AND attribute "foreigner" (all residents who are foreigners must register) c) Registration is mandatory for {attribute "resident"} AND {attribute "foreigner" OR attribute "have children"} (all foreign resident must register; national residents who have children must register)	
RE-5	**Definition of the results of a registration** An analyst must be able to report/input in the rule engine, in clear language, rules defining what are the results of a registration. The result of a registration has a name (example: registration number, registration certificate, permit, license, etc.). To this name, the analyst must be able to associate a template/document (see RE-11). In some cases, some subjects of the registration will receive a different result. The analyst can define through determinants (or combination of determinants) the different categories of subjects and can link each category of subjects to a specific result.	Must have

REQ-#	Requirement	Type & UC
	Examples: a) Future truck drivers that apply to a driving license and pass the exam will receive a driving license for "large vehicles", while car drivers will receive a driving license for "light vehicles" (attributes: "truck driver" or "car driver") b) Enterprises with assets below US$5,000 will receive a "Cottage Industry Certificate"; enterprises with assets above US$5,000 will receive a "Business License", when applying for an activity license (attributes: "assets below US$5,000" and "assets above US$5,000")	
RE-6	**Requirements of a registration - <u>Documents/Credentials</u>** An analyst must be able to report/input in the rule engine, in clear language, rules defining what are the documents (= credentials) that must be provided. ● A document/credential can be defined by a name ● One registration can have several required documents/credentials ● A document/credential can be physical or digital ● For each required document, the analyst can select one of the following options: has to be brought to the front desk; ○ has to be uploaded ○ has to be signed in front of an Operator. The required documents can differ according to the categories of subjects. The analyst can define, through determinants (or combination of determinants) the different categories of subjects and link them to specific required documents. Examples: a) Married applicants must provide a copy of their wedding certificate (attribute: "married") b) Foreigners must provide a copy of their residence permit (attribute: "foreigner") c) Applicants that can't provide a copy of their birth certificate must provide a certified copy of their ID (attribute: "can't provide a copy of the birth certificate")	Must have
RE-7	**Requirements of a registration - <u>Fees</u>**	Must have

REQ-#	Requirement	Type & UC
	An analyst must be able to report/input in the rule engine, in clear language, rules defining what are the fees of a registration. The fees of a registration has a name (example: State fee of MCTS programm, license registration fee etc.). To this name, the analyst must be able to associate an amount and a currency. ● Fee can be calculated or fixed; ● One registration may have 0..n fees; ● All fees must have an assigned currency. The system may allow multiple currencies; ● A fee can have a description. When the fee is calculated, a formula builder will allow the analyst to build the formula, using values from any field in the service and numeric values, combined with the usual operators (+, -, *, /). In some cases, some subjects of the registration will receive a different fee. The analyst can define through determinants (or combination of determinants) the different categories of subjects and can link each category of subjects to a specific fee. Examples: a) Mothers with one child that apply to a registration to MCTS program will receive a identity card for 10 EUR, while mothers with two children or more will receive the identity card for 15 EUR (attributes: “one child” or “more than one child”) b) Farmers with farming land area > 10 000 sqm that apply to registration of farm land will receive registration certificate for 10 EUR, while farmers with land area <= 10 000 sqm will receive registration result/credential certificate for 5 EUR.	
RE-8	**Requirements of a registration - Data/Claims** An analyst must be able to report/input in the rule engine, in clear language, rules defining what is the data (= claims) that must be provided. A piece of data is defined by: ● A name (e.g. date of birth, nationality, first name, last name, etc.) ● A type (text, number, date)	Must have

REQ-#	Requirement	Type & UC
	● mandatory/ optional (See Control configurator element for more options) As for the other requirements, some data can be required only for a specific category of subjects. The analyst can define, through determinants (or combination of determinants) the different categories of subjects and link them to specific required documents. Examples: a) Married applicants must provide the first name, last name and date of birth for their spouse (attribute: "married") b) Owners of farm land should provide the number and date of registration of their property; applicants who rent farm land must provide the name and ID number of the owner (attribute: "own land", "rent land")	
RE-9	**Possibility to combine various registrations in one service** In many cases the process for the user/applicant contains multiple pre-and post registration steps in order to achieve the final goal (e.g. apply for a healthcare program). For example, in order to register a mother and a newborn child to the Mother and Child support program, both of the subjects must be previously registered in the Civil/Population registry. Civil registry registration service could be a separate service, but it is much user friendlier and less time consuming for the applicant to merge the two registrations into one service that can be filled at the same time. This service type is called Single Window- or integrated registrations service. To facilitate the combination of various registrations in one service: ● Registration requirements (documents/credentials, results, fees) must be reusable inside the service. One registration may re-use another registration result credentials as input requirements. ● Overlapping requirements and/or results must be automatically eliminated. ● Fees of all registrations must be merged into one sum, so that the applicant could see and pay total fees. Fee information is stored and segregated per registration. In one registration the document/credential may be issued as a result and in another registration the same document/credential may be required as	Must have

REQ-#	Requirement	Type & UC
	an uploadable requirement. The system must eliminate the overlapping results and/or requirements if the requirements overlap inside the service. Examples: One service has two registrations and both of them require a passport to be uploaded. When an applicant chooses to apply for both registrations then the system must ask for the passport upload only once. One service has two registrations. First registration's result (credential) is the second registration's requirement. The system must not ask this requirement from the applicant as the result will be generated during the process. However, if the user chooses to register for only the second registration then the requirement must be asked.	
RE-10	**Creation and functioning of determinants** The analyst can define through determinants (or combination of determinants) the different categories of subjects and can link each category of subjects to a specific element of a service. An element of a service can be a field, block, button, message, processing role, result, requirement etc. Simplest case being, is this claim data field on the screen relevant for all subjects or only for a specific category of subjects? If all then no determinant is needed as all users will see and use it. If a specific (e.g. "applicant is married") category and this category must fill in additional information (e.g. "Spouse name"), then a determinant must be added to control the field visibility for this category of subjects. If the user is married, then the determinant is true and the "Spouse name" field is visible; if the determinant evaluates to false, then the spouse name field is not visible for the subject. Analyst must be able to: ● create and manage determinants (CRUD); ● search the existing determinant; ● reuse determinant inside the same service; ● use formulas - calculate math functions on the form with user input value and use the calculated value in the determinant. E.g. math functions (SUM, MIN, MAX, AVG, COUNT, MEDIAN, CEILING, FLOOR, ROUND, CONCAT, CONCAT_WS, LEN, REPLACE, UPPER_CASE, LOWER_CASE).	Must have

REQ-#	Requirement	Type & UC
	Apply determinants to: - Any element of a service (registration, claims/data fields, screens, roles, templates etc.) - Any element of a registration (results/credentials, requirements/documents) Determinants can be combined by "AND" and "OR" operators. Combinations can be grouped into "groups of determinants". Group of determinants can be combined through "AND" and "OR" operators.	
RE-11	**Definition of a template associated to the result of a registration** An analyst must be able to create an electronic template (a screen with images, text, fields, QR code) and to link it to the result of a registration. A template can be: - Filled with values coming from fields in the service or from external databases; - Printed as a pdf file. - Generated and uploaded to the system as a PDF file. On each template, an analyst can create data fields that must be filled by the system and place them on the screen of the template. Fields will have the following characteristics: - A name (e.g. date of birth, nationality, first name, last name, select option, submit application, etc.) - A type (text, number, date, selector, catalogue, button etc.) - Source of information/capture data from. Function to capture data from application screens upon creation of PDF. In addition to fields, the analyst can create information texts and add images, QR codes on the template screen. QR code must be generated with the ISO/IEC 18004:2015 standard. Determinants and groups of determinants created in the rule engine can be assigned to each field on a template. A field will be displayed/printed only if the determinants assigned to it are true. Fields can be grouped into containers (blocks, tables etc). A container is defined by a name.	Must have

REQ-#	Requirement	Type & UC
	Fields can be moved to and on the screen by “drag-and-drop, inside/outside of and between containers.	

6.1.2 User Interface and Flow Builder Functional Requirements

The purpose of the **Screen and flow builder** is to define and display the screens, and the fields on each screen, in the application file and processing parts, and to pre-fill or capture the data entered by the users of these screens in:

- The **application file**, where applicants provide claims (= fill a form), credentials (=upload files) and fees (= pay online or upload a payment receipt) and send his/her request to one or more entities in charge of the registration
- The **processing part**, where one or more human or automated (“robot” or “BOT”) operators can review the information (i.e. the data and documents) provided by the applicant, approve or reject an application, send claims to a registry and issue a credential

REQ-#	Requirement	Type & UC
SF-1	**Application file - Creation of screens and of their sequence** An analyst must be able to create one or more screens that will allow to show information to the applicant and to display fields that the applicant will have to fill to provide the requirements of the registration. The analyst can define the sequence/order in which the screens will be displayed to the applicant. To this end, the analyst will be able to activate or inactivate, through a toggle, the following screens: ● **Guide** screen: this is a screen where the analyst can create text for the information of the applicants (guidelines) or ask questions (through fields) to determine what category of subjects the applicant belongs to (i.e. if the applicant is subject to the registration and what data, documents and fees are required). ● **Applicant form** screen: this is where the applicant will provide the required data. ● **Documents upload** screen: where the applicant will upload scanned copies of the required documents/credentials.	Must have

REQ-#	Requirement	Type & UC
	● **Payment** screen: where the applicant will see a list of fees that must be paid and can be redirected to an external online payment service. ● **Send** screen: where the applicant can be reminded of any information missing in his/her application (if some fields have not been filled or documents not uploaded) and will be requested to confirm his/her will to apply for registration. The above screens, when activated, will be displayed, in the predefined order (guide, applicant form, document upload, payment, send). The flow of screens can be visualized by the analyst.	
SF-2	**Application file - Creation of fields on each screen** On each screen, an analyst can create data fields/claims that must be filled by the applicant and place them on the screen. Fields will have the following characteristics: ● A name (e.g. date of birth, nationality, first name, last name, select option, submit application, etc.) ● A type (text, number, date, selector, catalogue, button, etc.) ● The name of the registration(s) the field is associated with (i.e. the corresponding data is a requirement for this registration) ● Mandatory or optional Determinants and groups of determinants created in the rule engine can be assigned to each field. A field will be displayed only if the determinants assigned to it are true. Every time a field is created it is recorded in the “data” part of the rules engine. Fields can be grouped into containers (blocks). A container is defined by a name. Fields can be moved to and on the screen by “drag-and-drop, inside/outside of and between containers. In addition to fields, the analyst can create information texts, images that, contrary to fields, do not expect any input from the applicant.	Must have

REQ-#	Requirement	Type & UC
SF-3	**Processing part - Creation of screens/roles** The processing part relies on the flow builder because roles are part of the process flow. An analyst can create one or more screens allowing human operators to process the application files, i.e. to review the information sent by the applicant, to add data or documents to the application file (e.g. a number, a date, a credential, etc.), to approve an application, to reject it or to send it back to the applicant when more information is required. We call "processing role" (or simply "role") each successive processing an application file will go through until final approval is given and the registration is completed. Usually, different roles are ensured by different government entities. It happens that successive roles are ensured by different units in the same entity. Example flow: a) Unit 1 reviews the application file, checks that all data and documents provided seem true and correct b) Unit 2 approves the application c) Unit 3 issues and signs the credential and sends it to the applicant For each role, two screens are necessary: ● A first screen (**list screen**) where the operator will be able to: ○ see a list of application files, filtered by status (pending, validated, rejected, sent back to the applicant) ○ select a file for processing (if pending) or for consultation ● A second screen (**processing screen)** where the operator will be able to: ○ review the selected application file (data and documents provided by the applicant, fees paid or not) ○ add data or documents to the file ○ Make decision- approve the file, reject it or send it back to the applicant Therefore, an analyst must be able to create one or more roles, each role coming automatically with one list UI and one processing screen UI.	Must have

REQ-#	Requirement	Type & UC
	Each role will be defined with the following elements: ● Name of the role ● Name of the registration the role is associated with ● Entity in charge of the role ● Position in the process flow. Determinants and groups of determinants created in the rule engine can be assigned to each role. A role will be displayed/activated only if the determinants assigned to it are true. If Role determinants evaluates to false, then the application file passes the processing role without stopping.	
SF-4	**Processing part - Ordering of screens/roles in Flow Builder** An analyst will be able to define, for each role, a list of possible statuses, by selecting which of the following statuses are possible for the role: ● **Pending** (the file is waiting to be processed by the role) ● **Approved** (the file successfully passed the role) ● **Rejected** (the application is denied, the file is closed) ● **Send back to the applicant** for correction (the operator sends the file back to the applicant and requests that some information be corrected or added) For each activated status, the analyst will be able to indicate where the application file will be sent in the processing flow: ● To another processing role ● To the applicant role (for consultation or payment) ● To end- the application file is closed/ end of processing Therefore, the roles/screens in the processing part will come in an order specified by the analyst and this functionality is called Flow Builder. The analyst must be able to visualize the flow of roles/screens.	Must have
SF-5	**Processing part - creation of fields on processing screens** On each screen, an analyst can create fields that must be filled by the role operator and place them on the screen. Fields will have the following characteristics:	Must have

REQ-#	Requirement	Type & UC
	- A name (e.g. registration number, date of registration, type of registration, validate application, etc.) - A type (text, number, date, selector, catalogue, button, etc.) - The name of the registration(s) the field is associated with (i.e. the corresponding data is a requirement for this registration) - Mandatory or optional Every time a field is created it is recorded in the "data" part of the rules engine. Fields can be grouped into blocks, columns, field sets and other containers (tables). A container is defined by a name. Fields can be moved on the screen by "drag-and-drop, inside/outside of and between blocks. In addition to fields, the analyst can create information texts and images that, contrary to fields, do not expect any input from the applicant. Only human roles have the option to build processing screens. Determinants and groups of determinants created in the rule engine can be assigned to each field. A field will be displayed only if the determinants assigned to it are true.	
SF-6	**An analyst must be able to add QR-code/barcode scanning function to the service screen** An Analyst can add a form field element (e.g. button) to a screen and configure an action to trigger the capture of data from a QR-code. Expected result- Analyst will build a service that has a button on the screen and a field on the screen to store the captured data. An applicant/user can trigger the "Read QR code" button on the screen of a service that opens the mobile phone's (or other device) camera, reads the QR-code and captures the data from QR code to a field on the screen. Example data extracted from the QR-code is "MCTS123". QR code must be generated with the ISO/IEC 18004:2015 standard.	Must have UC2
SF-7	**Actions** - the analyst must be able to configure triggering of actions when a user or system initiates an event on a screen of a service. A triggering event can be a button click, a form element click, data being entered into a field, or a row being added to a table.	Must have UC1

REQ-#	Requirement	Type & UC
	The action attribute specifies one or more of the following data mapping (BOT): ● where to send the form-data when an event is triggered (Data BOT action) and where to store the received data (response). For example, the user will enter first name and last name and after entering the last information the system will initiate a data BOT action to search data from the external data source and store the answer(response) phone number to the correct data field; ● where to pull data (from an external registry API) and where to show the answer in the user interface fields; ● where to send user (screen flow action). For example, the button click will take a user to the next screen. The next screen can be internal screen or external URL; ● validate captured data integrity (required fields must be filled); ● save entered information; ● send message (screen message, sms, e-mail, API message etc.) Some actions can be activated only for a specific category of subjects. The analyst can define, through determinants (or combination of determinants) the different categories of subjects and link them to specific actions. Examples: a) User enters a person's identifier (tax number) and the button pulls first name, last name and date of birth from external API on the screen. b) Married applicants must provide the first name, last name and date of birth for their spouse (attribute: "married") and when data is entered, then the application screen will validate the entered persons name from the Population Registry and return validation answers on the screen (Success, Fail). Actions are only triggered if the user has entered the data AND "person is married"-determinant. c) User clicks the "Validate and send" button on the screen, then the system will activate three actions: Save user data action, validate user data action, send user data action. d) User clicks the "Print" button and the system triggers "Print to PDF" data action where a template is used as base for creating a new PDF document. System will enable the user to see, print or	

REQ-#	Requirement	Type & UC
	download the generated PDF. e) User submits application and system will trigger BOT Role in the flow and it in turn activates data action bot -e.g. POST/GET message to external API.	
SF-8	**Formulas -** Analysts must be able to add formula calculations to and between the fields. The Formulas can be added to numeric fields, int, decimal, date. Calculated values allow calculating values based on the values in other fields of the form. E.g. If the registration subject has more than 2 children, then multiply the social payment times the number of kids.	Must have UC2
SF-9	**Preview-** Analyst must be able to preview service User Interfaces before publishing the service to the applicants. Following preview functions must be available.: 1. Preview of user interfaces one by one; 2. Preview of full service; 3. Preview of full service in test instance with the functionality to test the full service before publishing to live instance. .	Must have UC1
SF-10	**Import/Export of service descriptions-** Analyst must be able to import/export full service description. Service descriptions must contain at least: 1. User interfaces, screens, fields; 2. Process flow; 3. Service settings. As a result, the service can be imported with minimal effort from another instance and published for users/applicants to use. Instance specific configurations must be done in each instance and are not target to import export.	Must have

6.1.3 Control Configurator's Functional Requirements

Control configurator, to check if the claims (claims and credentials of registers) are complete and true. This is done through:

- Define if the field is mandatory(required) or optional.
- Add input masks (e.g. only numbers, text , alphanumeric, etc)
- Checks between fields in the screens, web services with external databases, or human revision (by an operator of the entity in charge of the registration)

Pre-requirements:

- User is authenticated and authorized to use this administrative Control Configurator functionality.

REQ-#	Requirement	Type & UC
CC-1	**Controlling the data capturing- Form field value validation** An Analyst must be able to configure field specific validation options e.g.: ● required; ● number, bigger, smaller, min, max; ● text, regular expression, contains, mask; ● date, earlier, later, today, older than age (The age of the applicant must be >=18); ● File upload size max limit; ● File upload type allowed.	Must have UC1
CC-2	**Controlling the data capturing- Form field value validation from external API** An analyst must be able to configure screen field(s) validation from external API data source. Examples: ● Applicant name and ID must match with data in Civil Registry record; ● Subject (first name, last name, DoB) must not been entered to the MCTS registry.	Must have UC1
CC-3	**Controlling the data capturing- data integrity validation** The system must verify that all claims/fields are correctly captured and all required documents uploaded. The system must generate on screen meaningful error messages if any requirements are not fulfilled. The system should not let user to submit application file if screen data capturing is incomplete.	Must have UC1

6.2 Online Registration Services Functional Requirements

The purpose of the online registration services (from here on e-services) module of Registration BB is:

- to enable Applicants to apply for and receive registration claims(certificated documents);
- to enable the Back Office staff, i.e. Operators to process applications, register information and issue registration certificates.

Example UC: Maternal and Newborn Health USE CASE
Sona registers Sowmya in MCTS program. With Sowmya's consent, Sona registers Sowmya's child's name, address and birth certificate, and Sowmya's name and ID as the caretaker of the child into the MCTS system, which automatically validates the birth certificate and Sowmya's ID with the government's citizen records system. Sowmya then creates an account in MCTS for her electronic health records (EHRs) and a barcoded unique ID card for getting further assistance. MCTS connects Sowmya's mobile phone number to her ID and enables permission for Sona to electronically coordinate various MCTS services for Sowmya.

User story
As an Applicant I want to use an e-service, so that I can apply for multiple logically grouped registrations with one integrated service and receive all needed claims/certificates simultaneously.

As an Applicant I want to:

1. go through the filling process and submit an application to receive a registration (certificate);
2. apply for multiple registrations within a single service;
3. monitor the processing status of my application;
4. see the history of all applications submitted by myself.

Pre-requirements:

- User/applicant has access credentials;
- Electricity and internet is available.

Post-requirements:

- Applicants can log in to the system and see available e-services;
- Applicants can select relevant services and apply for a registration(s);
- Applicants can submit applications to relevant entities;

- Applicants can see applications in draft, rejected, validated and sent back for corrections status;
- Entity operators can see list of received applications and process the applications;
- Operators can make decisions (three types) and upload the decisions to the system as results;
- Operators can see statistics of the processings.
- Each back office operator can only see relevant data of the application. Operators are authorized to see and process their role related applications. .

REQ-#	Requirement	Type & UC
DS-1	**User Account (A0)** The User Account page, A0, is the first page that all users see when they authenticate (log in) to the system. A0 must contain the following sections: ● "My applications" part with various tabs: ● list of applications (can be filtered and queried by status); ● list of documents in the account and access to each of them; ● application file content - the data sent with the applications; ● messages received. ● "Services available" part listing the services available for applicant to use. For each service there is a button, an icon, a title, an explanation text. In order to make the service usable and personalized, Applicants must authenticate to the system - then the Applicant's information is automatically attached to the application file when sending the application for processing. The system must enable to: 1. see the User Account screen(Dashboard); 2. select an e-service from the list of e-services and fill a new application; 3. open the Applicant's own applications to see the content; 4. save and continue filling the Applicant's draft applications; 5. filter applications (search free text from application name, submission date, deadline date, and status); 6. delete draft applications; 7. see the content (Data, documents) of the submitted applications	Must have

REQ-#	Requirement	Type & UC
	and the status of the application; 8. see “My documents” - certificates and/or approvals issued for the Applicant; 9. see and give approval for applications waiting for the Applicant's approval; 10. see, correct and resubmit applications sent back for correction; 11. see applications rejected by the Back Office; 12. see messages sent for the Applicant; 13. see and CRUD all documents uploaded by the Applicant; 14. see a list of registration application files, filtered by status (pending, validated, rejected, sent back for correction); 15. send on screen messages and e-mail, sms, etc. messages to applicant(s) and users based on application statuses; 16. see the status of the application file of registration(s) process.	
DS-2	**e-Service screens** An applicant must have an option to see all screens of the e-service. The e-service may have one or more screens. Service screens are preconfigured in Screen and flow builder as service schema. Example screens: ● **Guide** screen: this is a screen where the analyst can create text for the information of the applicants (guidelines) or ask questions (through fields) to determine what category of subjects the applicant belongs to (i.e. if the applicant is subject to the registration and what data, documents and fees are required). ● **Applicant form** screen: this is where the applicant will provide the required data. ● **Documents upload** screen: where the applicant will upload scanned copies of the required documents/credentials. ● **Payment** screen: where the applicant will see a list of fees that must be paid and can be redirected to an external online payment service. ● **Send** screen: where the applicant can be reminded of any information missing in his/her application (if some fields have not been filled or documents not uploaded) and will be requested to confirm his/her will to apply for registration. The above screens, when activated, will be displayed, in the predefined order (guide, applicant form, document upload, payment, send) as a wizard. If not activated, then applicant will not see the pages.	Must have

REQ-#	Requirement	Type & UC
DS-3	**e-Service data capturing** When an applicant is entering data the system must capture data entered. Applicant can save their applicant file as a draft and continue data entering later.	Must have
DS-4	**e-Service data validation** When an applicant is entering data the system must validate the data based on configuration made by Analyst in the Control Configurator.	Must have
DS-5	**e-Service registrations** The system must tell applicants which services are applicable. ● The system must have the list of applicable registrations where the Applicant can make a selection before continuing to the next form page. A registration may be optional or mandatory. ● The system will not allow the Applicant to continue or submit the file if mandatory registrations are not selected. ● Applicants can select multiple registrations in the same application. Example: the UC-Postpartum infant care- Registrations in Civil registry, mother and child tracking program (MCTS) and optionally to pediatrician first meeting registration - three simultaneous registrations within one service.	Must have
DS-6	**e-Service required documents(requirements)** When an applicant has filled data/ selected registration to apply for, the system must show which additional information/documents must be uploaded with the application. Duplicate requirements are merged. If a requirement is an output of one registration and at the same time the input to the next registration, then this document must not be asked/visible in the requirements list as it will be acquired during the process. An Applicant must be able to upload documents or take the documents to the counter service as originals.	Must have
DS-7	**e-Service required fees/payments** When an applicant has filled data/ selected registration to apply for, the system must show which fees are relevant. ● Applicants must see the list of all fees to be paid, displayed both as sum and separately;	Must have

REQ-#	Requirement	Type & UC
	● Fees can be paid in bulk; ● Fees can be paid in the beginning of the process (prepeyment) ; in the middle of the process after a decision or after the process is finished upon receiving the result/output (cash payment); ● Payment options may include: online payment; cash payment; mobile payment, etc; ● All payment-related transactions must be logged; ● All payments received must be available in the payment registry (or equivalent registry) as successful transactions.	
DS-8	**e-Service movement on screens, roles and submission** Applicant can move between the screens and change the data up to the point of submission. Applicants can submit application for processing(flow). All submitted applications are recorded in processing flow engine and first role defined in the flow builder will receive the the registered application file as a task for processing. Movement on the screens can be done by clicking a button (Next) or by clicking on a tab/page name. Movement between the processing roles is done via decisions (approve, reject, send back) System must validate inconsistencies of the application upon submission.	Must have
DS-9	**e-Service application history** Applicant must see the application and flow history, process registration time and processing status in a flow. It should be possible to see which Institution is currently processing the application. Applicants must see the expected processing end time for each application.	Must have
DS-10	**Read information from QR-code/barcode and insert to the form** Applicant can use a function (e.g. button) and the action to capture data from QR-code. Expected result- applicant can activate (mobile phone) camera, read the QR-code and capture the data from QR code to a field. Example data to be captured: "MCTS31"; "www.registrations.org" <Example UI>	Must have UC2

REQ-#	Requirement	Type & UC
DS-11	**System audit log functionality** ● System logs all user action in the system. ● User action log is visible for admin users. ● By default user action log is stored for 1 year after which the system will delete the log. The storage length is configurable in the Rules Engine. ● Statistics are entered to statistics log table.	Must have UC3

REQ-#	Requirement	Type
DS-12	**Back office Operators must have a view to see the list of applications to be processed (task list dashboard).** ● The system must have the function for an Operator to pick assignments from the common task dashboard. ● The system must have the view for an Operator to see assigned roles and assigned tasks for this role. ● Operators are linked to Institutions (or sub-units) and roles, thus can only see the tasks relevant for their role and Institution. ● Optionally (configurable) - an operator could be able to claim a task from the application task dashboard. When task has been claimed, then the application file will be taken off from the common pool. <Example UI>	Must have
DS-13	**An Operator of a role must see the received application file screen containing all information submitted by an Applicant and information complemented by other Operators while processing the same file**. ● Required documents relevant for this role and registration linked to the role; ● Required data relevant for this role and registration(s) linked to this role. ● An Operator must not see any information that is not relevant/required for the registration that this role is serving. ● Must be able to see the history of the application file processing. ● Must be able to see the status of the application file. <Example UI>	Must have
DS-14	**An Operator must be able to make a decision in the system by selecting the right decision type (approve, reject, send back for correction).** ● At least one decision option must be available in order to process the application file. ● The system must enable a form for an Operator to draft a decision text and select/fill additional information on the form. ● In case of errors in the application, an operators must be able to mark which document and/or data field is incorrect. Applicants must see the highlighted information.	Must have

REQ-#	Requirement	Type
	<Example UI>	
DS-15	**Operators must have the option to print, sign and upload a certificate.** • The system must not let an Operator make a decision in case a required document/result certificate is missing. • The system can display, print and upload a filled certificate from a template.	Must have
DS-16	**Operators must have the option to edit application information if corrections are needed.** • The system must highlight if an Operator has made any changes to the information submitted by the Applicant. • The system should enable the Operator to remove and/or upload required documents.	Must have

6.3 Coverage Map

The coverage map shows how the **Functional Capabilities** (by an applicant and a registrar) in specific use cases match the functional requirements as described from page 10 in that document.
User Interface- Provide data in an online form, upload copies of credentials/claims/ documents.

Case management described in the following table is functionality description defined in the Dial use cases. In this document the used terminology is flow builder.

Use Case	User Journey	Functional Capabilities	Technical requirements of No-code development platform (6.1)	Technical requirements of Online registration services (6.2)
Registration	Postpartum and Infant Care	1. Capture Basic Details	Rules engine (6.1.1); User interface and flow builder (6.1.2); Control configurator (6.1.3);	6.2 Online registration services functional requirements: DS-1; DS-2; DS-3; DS-4; DS-5; DS-6; DS-8;
Registration	Postpartum and Infant Care	2. Optional- registration payment	Rules engine (6.1.1); User interface and flow builder (6.1.2).	DS-7
Registration	Postpartum	3. Optional:	User interface and	DS-2; DS-3; DS-4

	and Infant Care	Validate with foundational ID system	flow builder (6.1.2)	
Registration	Postpartum and Infant Care	4. Capture Registration Details	Rules engine (6.1.1); User interface and flow builder (6.1.2); Control configurator (6.1.3);	DS-3; DS-6; DS-5
Registration	Postpartum and Infant Care	5. Optional where legally required: Get Citizen Confirmation	User interface and flow builder (6.1.2);	DS-3; DS-6
Registration	Postpartum and Infant Care	6. Update Register. The local register is updated for later use.	Rules engine (6.1.1); User interface and flow builder (6.1.2); Control configurator (6.1.3);	DS-3; DS-5
Registration	Postpartum and Infant Care	7. Capture Card Details	User interface and flow builder (6.1.2); Control configurator (6.1.3);	DS-10
Registration	Postpartum and Infant Care	8. Assign Card to Parent	User interface and flow builder (6.1.2);	DS-3; DS-8; DS-5
Registration	Postpartum and Infant Care	9. Update Registry The card number and mothers details are sent to the Govn. Registry for update, e.g. Department of Health or Home Affairs.	User interface and flow builder (6.1.2);	DS-3; DS-5; DS-8
Registration	Postpartum and Infant Care	10. Capture Basic Details	Rules engine (6.1.1); User interface and flow builder (6.1.2); Control configurator (6.1.3);	DS-3; DS-5
Registration	Postpartum and Infant Care	11. Update Register The local register is updated for later use and	User interface and flow builder (6.1.2);	DS-3; DS-5; DS-8

		submit the information for the remote Govn. Department Registry.		
Registration	Postpartum and Infant Care	12. Request Birth Certificate	Rules engine (6.1.1); User interface and flow builder (6.1.2);	DS-3; DS-5; DS-8; DS-12; DS-13; DS-14; DS-15
Registration	Postpartum and Infant Care	13. Give Reference Number. Once the request is successfully committed the mother is given a reference number.	User interface and flow builder (6.1.2); SF-7	DS-1; DS-14; DS-15
Registration	Postpartum and Infant Care	14. Search Patient Case Generate a new folder for case records and link it to the child ID. If no case exists, create one and link the Card, otherwise identify the correct record and store the case UID.	User interface and flow builder (6.1.2); Control configurator (6.1.3); SF-7	DS-14; DS-15

Registration	Postpartum and Infant Care	15. Update Registry
		The card number, mothers details, childs details and birth details, are sent to the Govt, e.g. the Department of Home Affairs.
			Rules engine (6.1.1);
			User interface and flow builder (6.1.2);
			SF-7;
				5.2 The online registration services functional requirements;
				DS-3; DS-5; DS-8; DS-9; DS-12; DS-13; DS-14; DS-15

Payments	Postpartum and Infant Care	1.HC worker logs into the postpartum payment registry	User interface and flow builder (6.1.2);	DS-1

		system		
Payments	Postpartum and Infant Care	2.Mother presents program membership ID card to the HC worker, e.g. barcode	User interface and flow builder (6.1.2);	DS-10
Payments	Postpartum and Infant Care	3. Capture details for the Mother Capture data to verify the mother and prevent fraud per legal requirements, for processing later. API spec.	User interface and flow builder (6.1.2);	DS-3; DS-4
Payments	Postpartum and Infant Care	4.1 Validate the mother has completed all steps	User interface and flow builder (6.1.2); Control configurator (6.1.3);	DS-3; DS-4
Payments	Postpartum and Infant Care	4.2 Verify mother has no pending incentive voucher for this milestone?	User interface and flow builder (6.1.2); Control configurator (6.1.3);	DS-3; DS-4
Payments	Postpartum and Infant Care	4.3 Ask HC worker to certify that all steps have been completed	User interface and flow builder (6.1.2); Control configurator (6.1.3);	DS-3; DS-4
Payments	Postpartum and Infant Care	5. Determine payment amounts for HC worker and mother	Rules engine (6.1.1);	DS-3; DS-4
Payments	Postpartum and Infant Care	6. For the mother, a cash payment is given via a paper payment voucher	User interface and flow builder (6.1.2); Control configurator (6.1.3);	DS-3; DS-4
Payments	Postpartum and Infant Care	9. Record payment status	User interface and flow builder (6.1.2);	DS-3; DS-5; DS-8; DS-9; DS-12; DS-13;

		and amounts in registry	Control configurator (6.1.3);	DS-14; DS-15
Case Management	Postpartum and Infant Care	1. HC worker logs into the MCTS registration system	User interface and flow builder (6.1.2)	DS-1;
Case Management	Postpartum and Infant Care	2. Caretaker presents program membership ID card to the HC worker	Rules engine (6.1.1); User interface and flow builder (6.1.2)	DS-10
Case Management	Postpartum and Infant Care	3. HC workers have access to minimal required data for the purposes of completing this process.	User interface and flow builder (6.1.2)	DS-5; DS-3; DS-4
Case Management	Postpartum and Infant Care	4. The HC worker may read information from the child's health records registry.	User interface and flow builder (6.1.2)	DS-3; DS-4
Case Management	Postpartum and Infant Care	5. The HC worker updates prescriptions for medication API spec.	Rules engine (6.1.1); User interface and flow builder (6.1.2)	DS-2; DS-3; DS-4; DS-5
Case Management	Postpartum and Infant Care	6. The HC worker updates prescriptions for tests	User interface and flow builder (6.1.2)	DS-2; DS-3; DS-4; DS-5, DS-8
Case Management	Postpartum and Infant Care	7. The HC worker updates prescriptions for nutrition	User interface and flow builder (6.1.2)	DS-2; DS-3; DS-4; DS-5, DS-8
Case Management	Postpartum and Infant Care	8. The HC worker updates prescriptions for immunization	User interface and flow builder (6.1.2)	DS-2; DS-3; DS-4; DS-5, DS-8

Case Management	Postpartum and Infant Care	9. The HC worker updates prescriptions for therapy	User interface and flow builder (6.1.2)	DS-2; DS-3; DS-4; DS-5, DS-8
Registration	Unconditional Social Cash Transfer	1. The admin logs in	User interface and flow builder (6.1.2)	DS-1
Registration	Unconditional Social Cash Transfer	2. The admin searches the Registration	User interface and flow builder (6.1.2)	DS-2; DS-3; DS-5
Registration	Unconditional Social Cash Transfer	3. The admin may create a new registration record if none exists for the potential beneficiary	Rules engine (6.1.1); User interface and flow builder (6.1.2) Control configurator (6.1.3);	DS-2; DS-3; DS-4; DS-5, DS-8
Registration	Unconditional Social Cash Transfer	4. The admin verifies the potential beneficiary is correct based on details provided by the potential beneficiary	Rules engine (6.1.1); User interface and flow builder (6.1.2)	DS-4;
Registration	Unconditional Social Cash Transfer	5. During enrolment, further data can be collected by the admin	Rules engine (6.1.1); User interface and flow builder (6.1.2)	DS-3
Registration	Unconditional Social Cash Transfer	6. Optional: Admin issues a family program membership ID card, e.g. barcode	Rules engine (6.1.1); User interface and flow builder (6.1.2)	DS-8
Registration	Unconditional Social Cash Transfer	7. Optional: Programme specific data is often entered into a separate Beneficiary Registry associated with a	Rules engine (6.1.1); User interface and flow builder (6.1.2)	DS-5

		Beneficiary Operations Management System (BOMS)*		
Full process.	UC-E-USCT-001: Eligibility Determination and Benefit Package(s) Design - Unconditional Social Cash Transfer	User Interface; Activate API	Rules engine (6.1.1); User interface and flow builder (6.1.2) Control configurator (6.1.3);	DS-2; DS-3; DS-4; DS-5, DS-8

6.4 Overarching Requirements

This section describes Registration BB requirements that must be fulfilled in all components.

REQ-#	Requirement	Type & UC
REQ-2	System audit log functionality ● System must log all user activity in the system. ● User action log is visible for admin users. ● By default user action log is stored for 1 year after which the system will delete the log automatically. The storage length is configurable in rules engine. **IDENTIFY**: Each building block MUST implement access and authorization audit, logging, tracing and tracking with alerts (minimally proxied or implemented through the API Management and Gateway services). See detailed Audit logging requirements in the Security Requirements specification.	Must have
REQ-3	Each building block MUST implement the ability to provision, deprovision and manage Identities and access rights (this may or may not be centralized for the whole architecture as a unified provisioning process). See the Security Requirements specification.	Must have

7 Data Structures

The resource model shows the relationship between data objects that are used by this Building Block.

7.1 Standards/Protocols

The following standards are applicable to data structures in the registration building block:

1. All dates should follow ISO 8601.
2. RFC 7159 - The JavaScript Object Notation (JSON)
3. Open -API Version 3.1.0
4. QR code must be generated with the ISO/IEC 18004:2015 standard

7.2 Resource Model

The resource model provides a basic description of each data object that is used by the building block. It also shows the relationship between data objects.

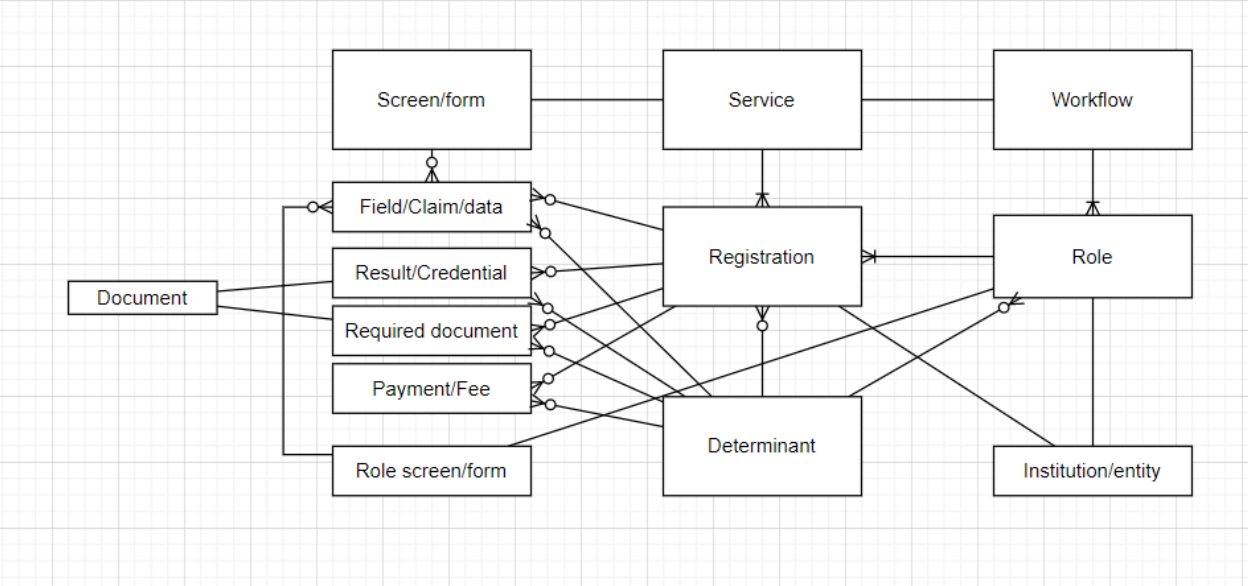


Illustration 4 - Resource model. See editable diagram here

7.3 Data Elements

The data elements provide detail for the resource model defined above. This section will list the core/required fields for each resource. Note that the data elements can be extended for a particular use case, but they must always contain at least the fields defined here.

Data Element	Description	Required
Registration	Process through which an entity gets claims recorded in a registry and a credential proving the registration, in exchange of providing some requirements.	yes
Service	Name given to a registration, or to a combination of registrations which can be undertaken simultaneously, by the entity(ies) in charge of organizing the registration process.	yes
Screen/form	A service is composed of one or several screens defined in the Screen Builder by the analyst.	yes
Role screen/form	One or several screens defined in the Screen builder. Screen(s) where a human operator will undertake actions to process an applicant file, usually enter data (eg registration number, or registration certificate) and press buttons (approve, send back for correction, etc.).	yes
Workflow	A workflow provides a visual representation of a business process. Workflow Engine executes processes that are defined in Business Process Model and Notation (BPMN), the global standard for process modeling. With BPMN, the analyst can automate most complex business processes using an easy-to-adopt visual modeling language.	no
Role	Human or bot actor in the workflow who must process the application file in the order set in the workflow. A role is defined by 4 properties: ● Name of the role and; ● Who is in charge of the role- Institution entity ● Type- Action that will take place in the context of the role, either human or BOT ● Status decision options of the application file in relation with the role (0=pending, 1=passed successfully, 2=send back for correction, 3=rejected)	no
Institution /entity	Institution in charge of Registration and/or Role.	yes
Field/claim/data	An attribute asserted by an entity, about itself or another entity	no
Result/credential	The result of a registration, in addition to the	no

	recording of information in a registry, is usually a credential (sometimes called: certificate, license, permit, card, etc.) proving the registration.	
Required document	Information (i.e. claims and credentials) which must be provided in a registration process. The requirements may vary according to each subject.	no
Payment/fee	Amount of money to be paid in relation to a registration.	no
Determinant	A determinant is an attribute, defined in the rule, which determines/triggers if (1) an entity is subject to a registration and/or (2) what requirements this entity must provide to register.	no

8 Service APIs

This section describes external APIs that must be implemented by the building block. Additional APIs may be implemented by the building block (all APIs must adhere to the standards and protocols defined), but the listed APIs define a minimal set that must be provided by any implementation.

Registration BB is expected to host the following API services.

8.1 Statistics API

The statistics API gives BB operational statistics- number of processed applications (per: operator, registration, service, date).

Resource model:

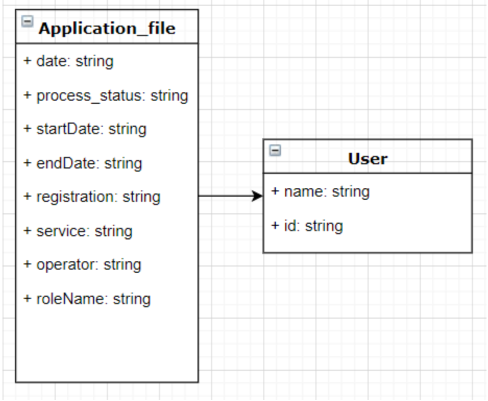


Illustration 5- resource model for Registration BB Statistics APi.

API specifications are described in the Github file.

Standards:

This section defines standards and protocols used by the API. The API is built using representational state transfer (REST) software architectural style (https://restfulapi.net/) and described in Open API 3 standard (https://swagger.io/specification/) using YAML (a human-readable data-serialization language - http://yaml.org/). Request and response body is in JSON (lightweight data-interchange format - https://www.json.org/json-en.html).

9 Workflows

This workflow section provides a detailed view of how this building block will interact with other building blocks to support common use cases. It lists workflows that this building block must support. Other workflows may be implemented in addition to those listed.

In this section two main workflows are described:

- Creating a registration service

- Using a registration service

For each workflow interaction, a sub-section is added.

9.1 Workflow- Creating a Registration Service (Admin Function)

9.1.1 Description

Analyst/administrator as the main actor in this workflow will create a new e-service by filling the required registration requirements, create screens for the user and publish the service on the web ready to be used by internet users.

As a pre-condition the user/analyst has authentication credentials and authorization to the service builder- Rules engine. If the service needs to read or write data from external sources (Information Mediator) that requires authorization then this is already authorized for the Registration BB. Registration BB is connected to Information Mediator.

As a post-condition, the e-Service has beglossaryen published on the internet for users to use immediately. The optimal time for an Analyst to build the e-service of a simple registration service is 2h.

9.1.2 User Journey

User journey is the step by step journey that a user takes to reach their goal. This journey consists of a number of website pages/screens and decision points that carry the user from one step to another. The user journey is used to map out the current journey a typical user might take to reach their goal

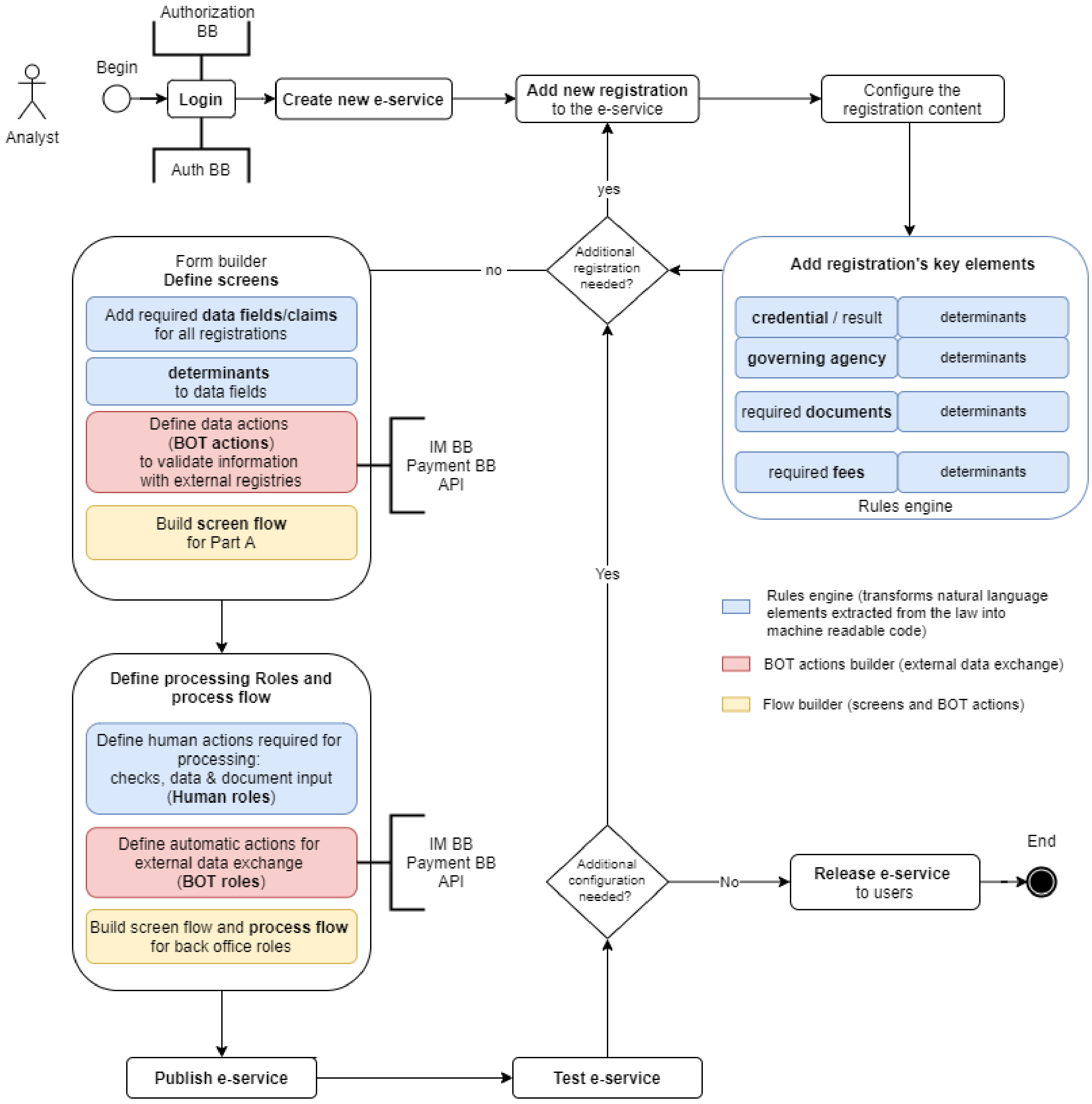

Authorization BB
Begin
Analyst
Login
Auth BB
Create new e-service
Add new registration to the e-service
Configure the registration content
yes
Additional registration needed?
no
Form builder
Define screens
Add required data fields/claims for all registrations
determinants to data fields
Define data actions (BOT actions) to validate information with external registries
Build screen flow for Part A
IM BB Payment BB API
Add registration's key elements
credential / result
determinants
governing agency
determinants
required documents
determinants
required fees
determinants
Rules engine
Yes
🟦 Rules engine (transforms natural language elements extracted from the law into machine readable code)
🟥 BOT actions builder (external data exchange)
🟨 Flow builder (screens and BOT actions)
Define processing Roles and process flow
Define human actions required for processing: checks, data & document input (Human roles)
Define automatic actions for external data exchange (BOT roles)
Build screen flow and process flow for back office roles
IM BB Payment BB API
Additional configuration needed?
No
Release e-service to users
End
Publish e-service
Test e-service

Illustration 6- user journey of creating a registration service. See editable file here.

9.1.3 Interaction with Other Building Blocks

This workflow requires interaction with following building blocks:

1. Authentication and authorization (Security BB)
2. User registration (Security BB)
3. Information Mediator (IM) BB
4. Payment BB
5. Setup for multiple Registration BB-s

For each interaction, the following information is provided:

- Name
- Sequence diagram
- Notes
- Data structures (link to data resources defined above)

Interaction 1: Authentication and Authorization (Security BB)

Name: Authentication - Existing user authentication (Security BB)

Sequence Diagram:

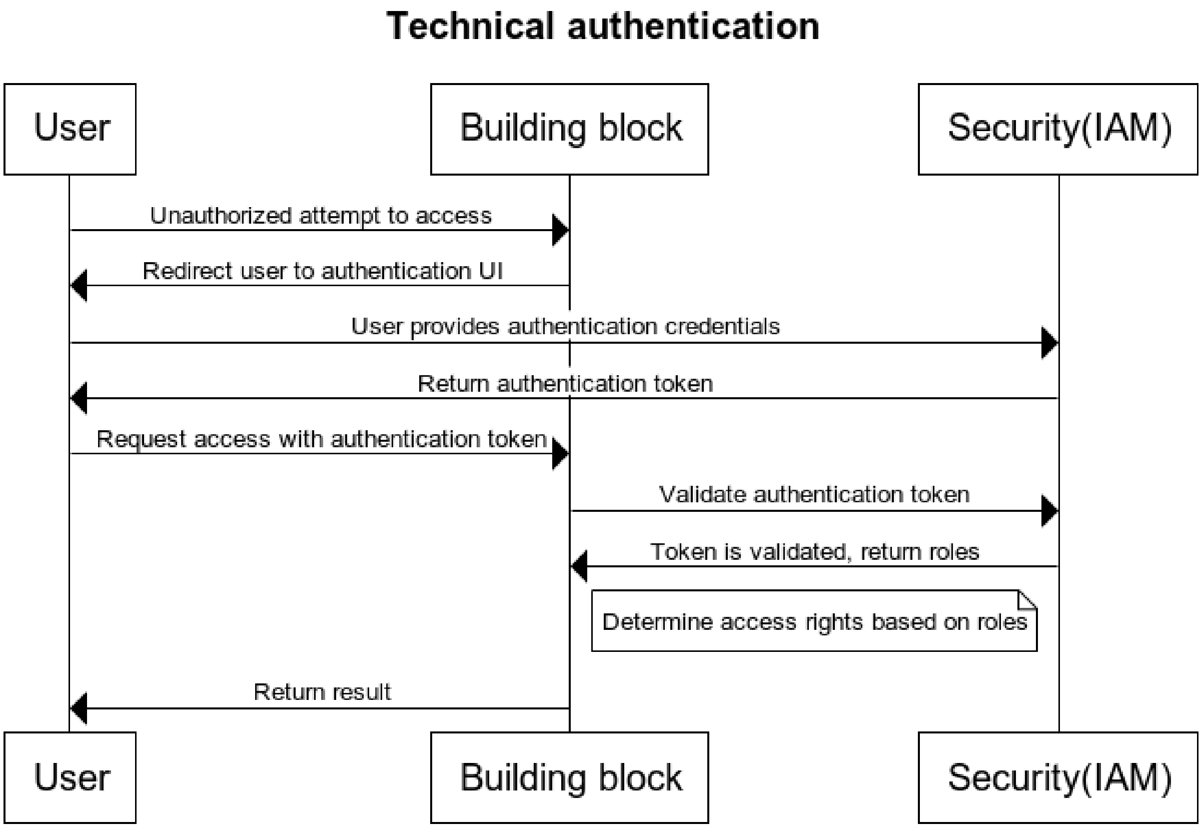

Technical authentication
User
Building block
Security(IAM)
Unauthorized attempt to access
Redirect user to authentication UI
User provides authentication credentials
Return authentication token
Request access with authentication token
Validate authentication token
Token is validated, return roles
Determine access rights based on roles
Return result
User
Building block
Security(IAM)

Illustration 7 - Authentication and authorization. See editable diagram.

Notes:

Pre-requirements: user has passed the provisioning of user credentials and this can authenticate/log in to Govstack (sandbox) system. User roles are also added by Security (IAM) system when token is sent to the Building Block. Registration BB in this case is operating as Building Block.
See full description of the user authentication in Security BB documentation.

Data structures:

Name	Required Data	Notes
Existing user authentication	Credentials: username/e-mail/ UID, Password, URL of client system.	User credentials vary depending on the country. Some countries may have additional user credentials, for example User ID number. Precondition- user is registered in the Govstack sandbox.
Authorization	Roles Example: “Service A”; “Registration Role A”; “Institution A”; “Part A”; “Part B”; “SADMIN”	IAM integration. Precondition- user is registered in the Govstack sandbox (IAM).

Interaction 2: User Registration (Security BB)

Name: User first time registration (Security BB)

Sequence Diagram via Self-Registration:

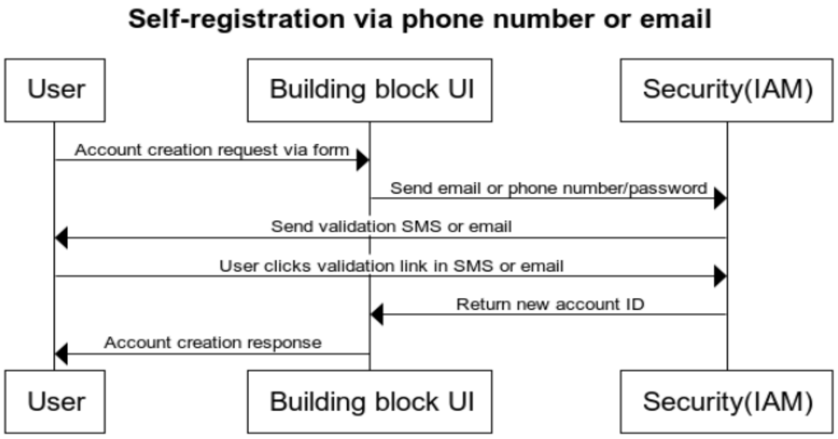

Self-registration via phone number or email
User
Building block UI
Security(IAM)
Account creation request via form
Send email or phone number/password
Send validation SMS or email
User clicks validation link in SMS or email
Return new account ID
Account creation response
User
Building block UI
Security(IAM)

Illustration 8- Self registration See editable diagram.

Notes:

Pre-requirements: user does not have user credentials to authenticate/log in to Govstack (sandbox) system.

Registration BB in this case is operating as Building Block UI.

See full description of the User registration in Security BB documentation.

Sequence Diagram for User Self Registration via Foundational ID:

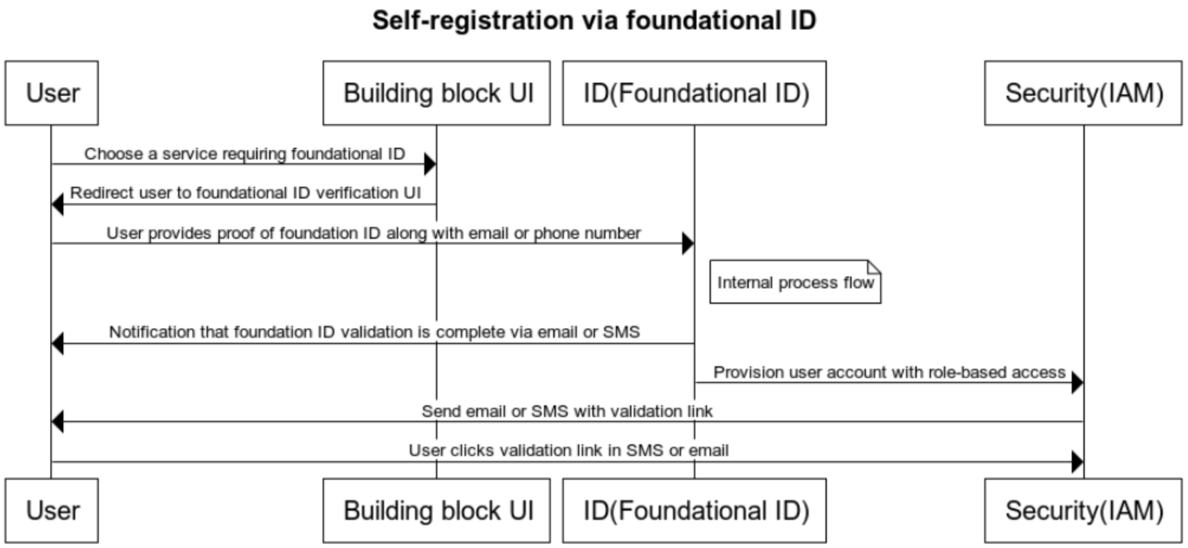

Self-registration via foundational ID
User
Building block UI
ID(Foundational ID)
Security(IAM)
Choose a service requiring foundational ID
Redirect user to foundational ID verification UI
User provides proof of foundation ID along with email or phone number
Internal process flow
Notification that foundation ID validation is complete via email or SMS
Provision user account with role-based access
Send email or SMS with validation link
User clicks validation link in SMS or email
User
Building block UI
ID(Foundational ID)
Security(IAM)

Illustration 9- Self registration via foundational ID. See an editable diagram.

Notes:

Pre-requirements: user does not have user credentials to authenticate/log in to Govstack (sandbox) system. Registration BB in this case is operating as Building Block UI. See full description of the User registration in Security BB documentation.

Interaction 3: Information Mediator (IM) BB

Name: Request API descriptions (Information Mediator BB)

Sequence diagram:

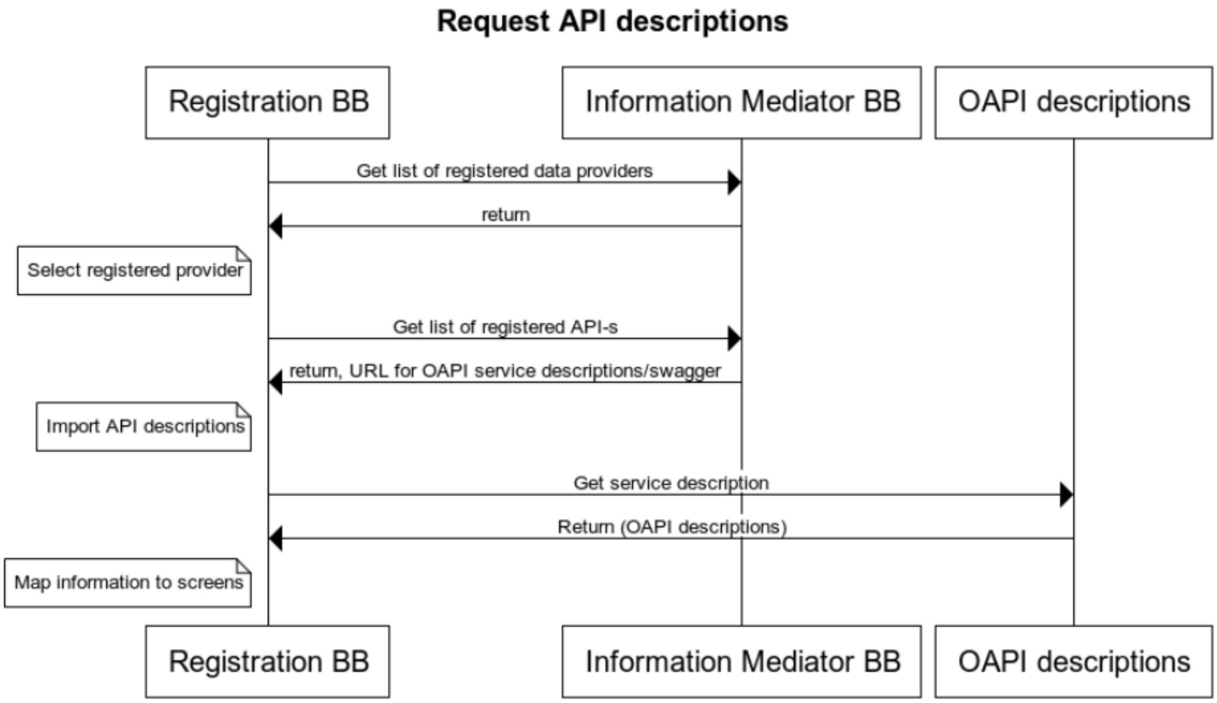

Request API descriptions
Registration BB
Information Mediator BB
OAPI descriptions
Get list of registered data providers
return
Select registered provider
Get list of registered API-s
return, URL for OAPI service descriptions/swagger
Import API descriptions
Get service description
Return (OAPI descriptions)
Map information to screens
Registration BB
Information Mediator BB
OAPI descriptions

Illustration 10- IM API description. See Websequencediagrams for an editable diagram.

Data structures:

Name	Required Data	Notes
Get list of registered institutions/databases	Return: List of data providers (institution name, system name, ID).	The returned list (institution ID, system name) will be given as an input to the next call to see the list of APIs.
Request API descriptions	Datasource name, ID, URL of the service and url for the service descriptions (swagger).	Pre requirements: *Subscription for Registration BB to use the IM system is done. *Before the Registration BB can use API information for screen populating with data the system must import API service

		descriptions.
Get service descriptions	Input: URL of the OAPI service. Output: OAPI descriptions	It works like Swagger. This endpoint may be external. Example API services to integrate to Registration BB screens: 1. Registry BB - Register of Mother and Child Program 2. Payment BB API to read received payments data.

Interaction 4: Payment BB

Name: Get list of payment options (Payment BB)

Sequence diagram:

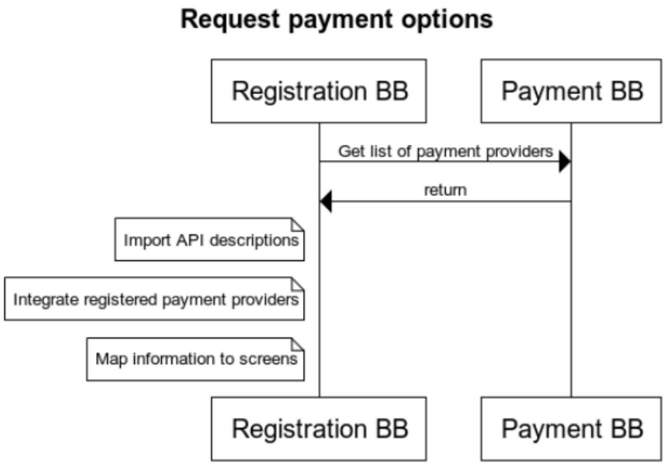


Illustration 11- Request payment options. See https://app.diagrams.net/ for an editable diagram.

Data structures:

Name	Required Data	Notes
Get list of payment options	Payment API descriptions, List of payment options, details, data, url.	Before the Registration BB can use the Payment BB information for screen populating the system must import API service

		descriptions.

Interaction 5: Setup for Multiple Registration BBs

Name: Setup for multiple Registration BBs

Description :
The Registration BB can be set up as a standalone BB facilitating multiple institutions and services in one system instance (See illustration 1 below).

Setup in one shared instance is required when one single window service is built together with multiple institutions (multiple registrations inside one service). This enables sharing of resources such as proceflow, screens, data, user rights, infrastructure, maintenance personnel etc.

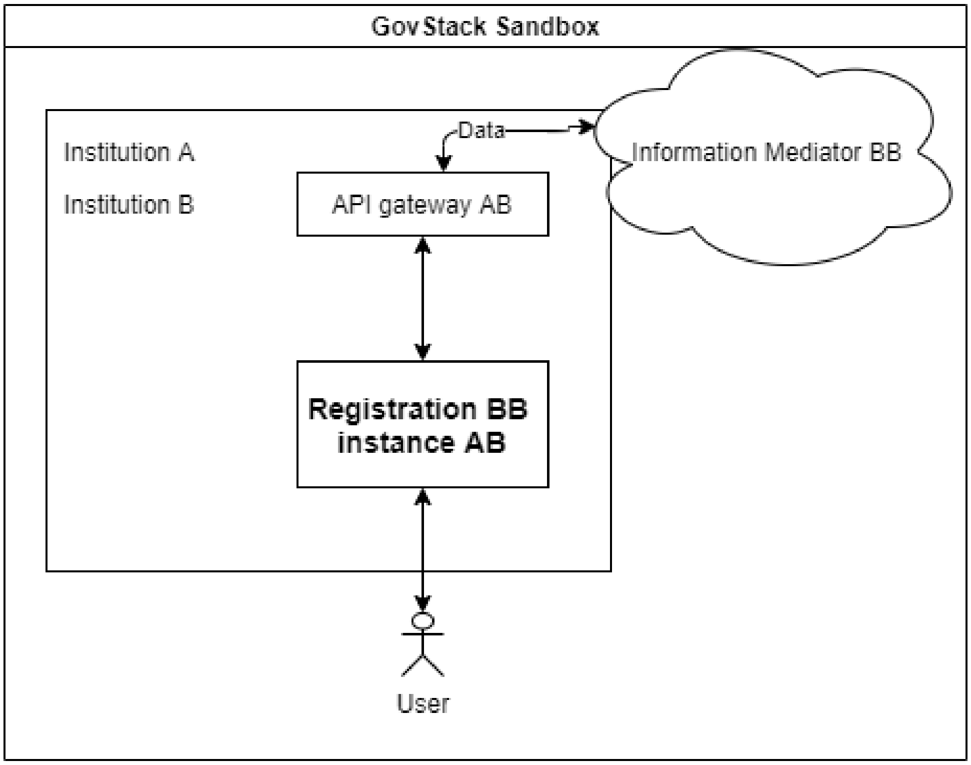


Illustration 14 - Domain specific setup diagram for multiple Registration BB-s in one ecosystem. See Diagrams.net for an editable diagram.

In other cases the Registration BBs can be installed in different instances (domain specific instances- see illustration 2 below) and in this case the BBs are not sharing resources and are not communicating with the other Registration BB. In these standalone instances the BBs are communicating with Information Mediator BB and other BBs as any other BBs.

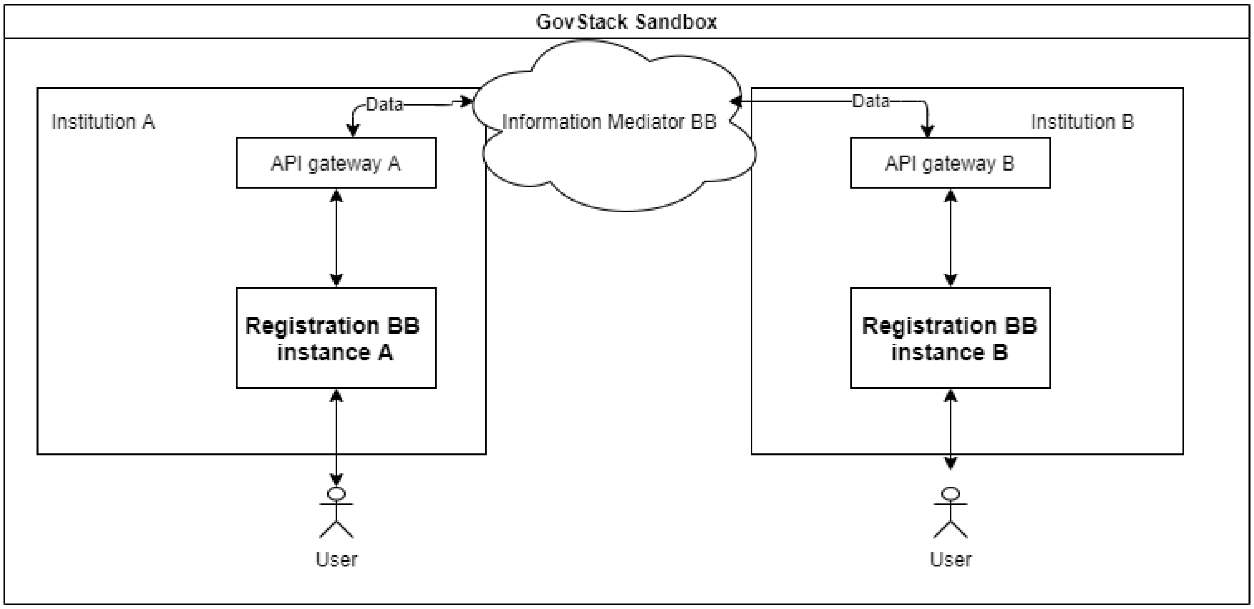


Illustration 15 - Domain specific setup diagram for multiple Registration BBs in one ecosystem. See Diagrams.net for an editable diagram.

9.2 Workflow- Using a Registration Service

9.2.1 Description

Citizen/ applicant, as the main actor in the process will open a web URL address and authenticate. Once authenticated, the applicant opens an e-service and fills required data fields /claims on the first screen. System/Rule engine will then validate if the user is applicable for any registrations and if so will open a list of mandatory or optional registrations for the user to select. Once selected the system will fill the form with required data fields relevant for the applicant to fill. In addition the system will show which documents are relevant for the applicant to upload and how much fees to pay. An applicant will then fill data, upload documents, select a payment method and make the payment (if required). If designed so the Control Configurator component will validate captured data against external API data and in case validation fails issue warnings to the applicant. After payment, the applicant can submit the application file for registration and processing. System validates if all required fields are correctly filled upon the submission.

Back office system registers all applications submitted. If configured so, the process flow may have bot(s) and human(s) in the flow. Bot roles process applications by validating information against external API information. When processed the BOT role will pass the processing task to the next role.

Back office operator as the main actor in the second part of the process receives a task notification message and authenticates to the web system. The system shows a list of applications assigned for the operator to process. Operator will then open the application and verify the application content and make a decision (approve, reject, send back for correction). If an application has discrepancies then the operator will mark the errors and/or explain the reasons for rejections. When the application file is positively processed by the back office operators, then the applicant will receive the credential/result of the application. If the back office processing flow has multiple roles then the approval by the first processing role operator will take the application processing task to the next role.

Preconditions:

As a pre-condition, the web system is accessible from the internet and service is published, the user/applicant has user credentials for authentication. No authorization is needed for applicants if the service is usable by all users.

The user/operator has authentication credentials and authorization to the web portal/back office system and to a service, if not open to all users. If a service needs to read or write data from an external source (IM-API) that requires authorization then this is already authorized for the Registration BB.

As a post condition, the applicant was able to fill a registration form and submit an application. As an operator, the user was able to see the pending applications and process the applications.

Sequence Diagram:

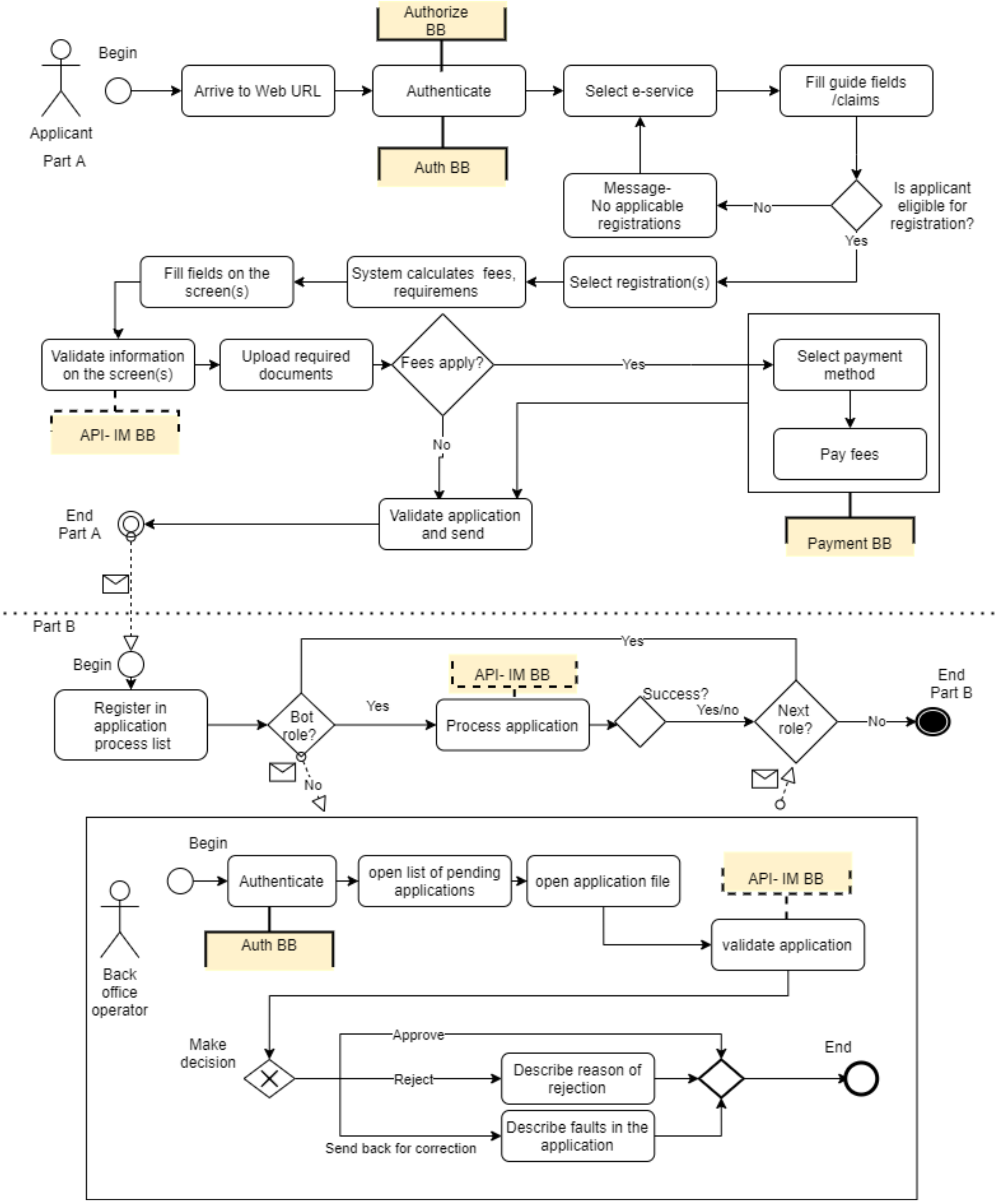


Illustration 12 - The process of filing an application file. See editable version here

9.2.2 Interaction with other Building Blocks

This workflow requires interaction with following building blocks:

- Authentication and authorization (Security BB)- see "Creating a registration service"
- Information Mediator (IM) BB
- Payment BB

For each interaction, the following information is provided:

- Name
- Sequence diagram
- Data structures (link to data resources defined above)
- Notes

Interaction 1: Voucher Pre-Activation (Payment BB)

Name: Voucher pre-activation (Payment BB)

Use case: e-service user initiates a registration service to issue a payment voucher for the subject/beneficiary.

Pre-conditions: Subject is identified and information is verified. User has credentials to authenticate in the Registration BB web interface. User has authorization to use the Voucher activation registration e-service. User has pre-activated vouchers printed on the paper. All communication is securely done via Information Mediator BB.

Post-conditions: A subject/ beneficiary will receive activated payment voucher and the voucher is activated in the respective external registry.

Sequence diagram:

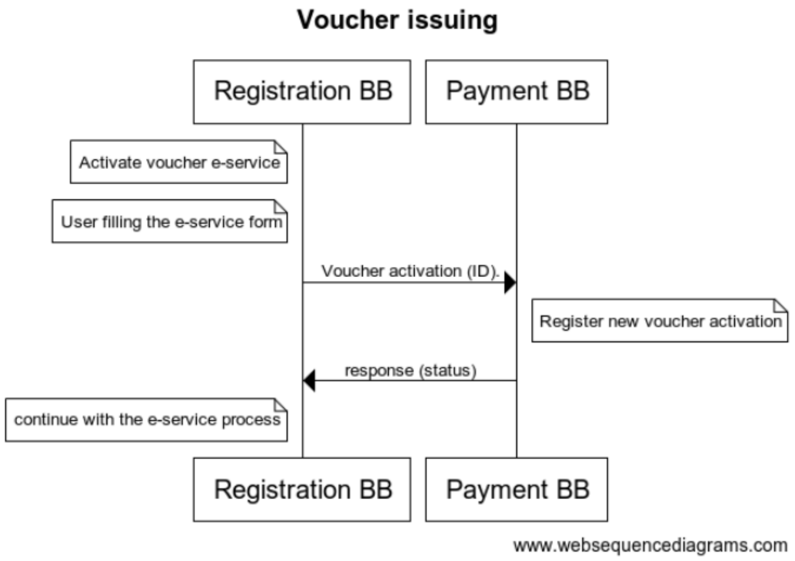

Voucher issuing
Registration BB
Payment BB
Activate voucher e-service
User filling the e-service form
Voucher activation (ID).
Register new voucher activation
response (status)
continue with the e-service process
Registration BB
Payment BB
www.websequencediagrams.com

Illustration 13 - Voucher issuing diagram. See https://app.diagrams.net/ for an editable diagram.

Data structures:

Name	Required Data	Notes
Voucher activation	Voucher ID	See more in Payment BB
Response	Status	The response is stored in Registration BB.

10 Other Resources

- API descriptions in Github
  https://github.com/ingmarvali/BuildingBlockAPI/blob/main/RegistrationBB/GovStack_Registration_BB_API_template-1.1.0-resolved.json

- Security requirements:
  https://docs.egovstack.net/v1.1.0/Security_Requirements_v1.1.0.pdf

- Architecture requirements:
  https://docs.egovstack.net/v1.1.0/Architecture_and_NonFunctional_Requirements_v1.1.0.pdf

- Information Mediator requirements:
  https://docs.egovstack.net/v1.1.0/Information_Mediator_Building_Block_Specification_v1.1.0.pdf

- Logical process blueprint:
  https://docs.google.com/document/d/1DRjpuyINjf6YVBRrEh9Q6VdB0zVzq1aqGQOukpktWZ8/edit#heading=h.z0zf4zjfif4c

- Infrastructure capabilities
  This depends on the amount of users, but optimal estimated hardware requirements are:
  - 8 Core CPU
  - 32GB RAM
  - 500GB SSD/HDD
  - 1000Mbit Network connectivity with dedicated publicly routable IPv4 address

System should be able to run either on VirtualMachine or on a dedicated server.

11 Key Decision Log

1. Registration BB and Digital Registries BB is separated into two blocks.

2. The UNCTAD's eRegistrations system will be used as a reference system as it fulfills most of the BB requirements.

3. Additional section "Coverage map" will be added.

4. “Coverage map” was moved to section 6.3.

5. Infrastructure capabilities added (9.5). 05.12.2021

6. Review comments incorporated to the main document 24.02.2022 (see below). Future consideration chapter updated based on reviewers comments. Recommendations not to be considered in this building block documented (see below)

1. Incorporated in V1	
Display, on one or various screens, the necessary fields for the applicant to enter the required data and documents (including a receipt of payment, in case the system does not propose an online payment option or if the user for some reason wants to make the payment physically or directly / in the traditional way).	Text changes added.
9.1 Workflow. such user journey implies ability to test defined forms and workflows, which means that there should test environments and capability to export/import configuration data between different environments. I am not sure that I saw requirements for that in the specification.	Yes, there must be an option to preview the service and to import export service descriptions. Two new requirements added: SF-9; SF-10 In addition there is an optional requirement to enable publishing of services into multiple instances (RE-2).
Also, vocabulary is not consistent. e.g. - Is 'Workflow' the same as 'Case Management'?	Consistency is improved. “Case management” is not a terminology used in this document, it is used to reference to the Dial use cases. Workflow term refers to building block internal workflows (user workflows) and not to the functionality. The term used in the document for functionality of workflow is Flow Builder, roles and Flow
Missing pieces - Queries/ Status/ Monitoring mechanism - How can a citizen query his/her status on the registration process hasn't been specified.	Status/monitoring requirements was added.
Citizen experience due to multiple statuses across departments and applications - If there are multiple processes for multiple applications, the citizens can potentially have many different statuses leading to suboptimal citizen	Some clarifications made to Requirement DS-1. in general the registration and application status visibility is described.

experience. How can these issues be handled using the specifications provided?	
Registration BB is focusing only onto getting data and not onto using Registry.	The Registration BB focuses both in capturing data and using registry data in the processes and sending it to external API endpoints. We will review the requirement to pull data to the form and rephrase so that it would be more understandable.

3. Not to be considered	
If 2 vendors implement a building block, will the 2 different implementation be interoperable.	Yes, all BB-s are interoperable via Information mediator BB. It is mandatory to use the Open Api standard for communication.
My comment more regarding the first sentence: „A registration may involve more than two parties: one or more third parties can be requested to assert/confirm the information provided by the applicant (a notary, a family member, a witness, another public entity, or a non-human entity such as a database);" because the issuer of the electronic signature /certificate guarantees the identity of the owner of the certificate and in that sense also guarantees financially that the owner of the signature /certificate is the person or entity for which he / she represents himself /herself. In that sense, it belongs to the "third parties" if I understood the meaning of this section well.	Modifications not feasible. Interesting thought to validate the uploaded documents signed with electronic signature. The purpose for this chapter was to validate the information filled in by the user (e.g. first name, last name, ID, DoB, etc.). I feel this electronic signature validation feature needs a use case, then we can see how we can describe the required functionality.
* Ensure that there is tracking of all changes to records ensuring that the Registry is a System of Record * Publish appropriate (i.e. anonymised) Open Data automatically on registration, and regularly (e.g. monthly) pseudonymised aggregate data	Already existing. First of all logging is described in the following requirements (REQ-2). Changes to records are done with API-s in Digital Registries BB (or existing BB) , thus logging and tracking of changes to records is available in another BB. Same for Open data, it will be published via Digital Registries BB (will be added to the next scope). Analyst decides which data field will be public and which is not.

3. Not to be considered	
Digital archiving capability is an absolute MUST capability for majority of registrations related to property, for example. It should be at least in the Roadmap for the Registration BB.	This is infrastructure implementation time requirement not BB requirement. Registration BB is like a proxy that stores data only temporarily. Once the processing is done the information will be moved to other endpoints where it will be stored. This is why long term storing is out for now. The Digital Registries BB (or existing registry) is an option to store the information for the long term.
In practice, there are very different types of user interfaces: (1) Back Office UI for couple of officers, for whom performance and convenience are most important requirements, (2) Self-Service Portal for individuals, who only occasionally using governmental services - for them simplicity and clarity are most important requirements, (3) Self-Service apps for professionals (for example, tax consultants, accountants, logistics etc) - they need something in between, perhaps. This section is targeting what kind of users?	System is flexible to build these self-service services and also processing screens in Back Office processing. The Registration BB could operate as Public Portal or Registry management system (for Back Office). Next paragraphs (6.1.2) describe the UI sections and the target users- Application File, Processing part. The Registrations BB is envisioned to be flexible to configure the user interface based on the user needs. In addition, the current requirements in this document are focused on the first target Use Cases. The personas are- Healthcare worker, parent, etc.
5.2 Registration is always done for some reason, which means that there are some third parties, which may want to have an access to results of registration. For example, if parent have granted permission to her child to cross a boarder and to go for sport competition abroad, then boarder guard officer should be able to see this permission electronically. I do not see such requirement here.	Content sharing agreement is dealt with Consent management BB.
5.2 In big cities, there are may be many places, where registration can be done (if physical attendance is required). In such cases there should be capability to book appointment and preliminary upload some contextual data.	Yes, the No-code development platform has been designed to be flexible enough to let the user to select a Place of Collection on the form, time of collection, and if required upload some preliminary information- e.g. passport copy. It has been decided that time reservation and Calendar function is solved in another BB.
DS-7 I would suggest explicitly distinguish two types of mobile devices: smartphones, which have 3G-4G and GUI and old-fashioned mobiles with USSD. Those USSD devices should be also supported - in Africa it is massively the only option	Modification is unacceptable. This is part of the Payment BB scope.

3. Not to be considered	
6. Standards. 1. As you refer here to rules and decisions, then I would recommend to utilise Decision Model and Notation spec (at least for so called Rule engine). 2. As you refer here to workflows, then I would recommend to utilise BPMN spec at least. 3. Also, CMMN might be interesting	BB is open to any of these standards and this is the decision of build time for developers. 1. The Rules engine is designed to be user friendly and it follows the well known Decision model- IF this is true then show the form element, requirement, any element in the system. It is not restricted to this decision model, so the developer can introduce other business rule methods. 2. BPMN may be technically ok, but overkill for this function in UI. Any user friendly logic will do. 3. CMMN Technically OK. user interface wise something simpler is needed.
6. Resource model This model seems to be limited only with the processing part of the registration and excludes usage of Registry part. Also, Digital Registries component doesn't show anywhere Registry-Subject model as per terminology section #2 (see above). Aim of registration process is always to create registry which can be heavily used for queries. For example financial services authority after issuing licenses will be doing regular risks monitoring and for that they need to have Registry of Companies with types of issued licenses with all related individuals. I am not sure that I understand how such usage can be implemented by the the Registration BB and Digital registries BB.	May be related to requirement SF-7. Registration BB is to collect and submit data to the registration Authority. Once the authority users have made a decision to enter data to the registry, then the system passes the data to Digital Registries BB (or to existing registry). If there are consumers of this registry data, then they have an option to ask this information via UI services in Registration BB or via Open API in Digital Registries BB.
6. such generic approach to a data model of the registration process seems to be not very practical. In reality, as terminology section put forward, there are Subject and Applicant, which may have its own complex structure. Omitting Subject and Applicant in the data model put the whole heavy load of modelling of the processing flow and Registry onto users and as such added value of such component seems to be too small. In reality, Subject always has some legal entities and/or persons and/or property/assets. And data attributes are normally attached to some combination of this triad: legal entity, individual and asset. I think, Registration BB and Digital Registries BB should address this complexity. Otherwise it will not be useful for public sector. It will be easier just to ask	We have chosen the principles of NO-CODE. Thus analyst must be able achieve the same goal without writing code. Data is defined based on domain in focus. Each e-service domain model is different. In this version our focus was postpartum infant care use cases. As the BB is highly flexible there is no need to write the entity data into the data model. Each process has its own data model and the system must be able to facilitate this (all domains). In Mother and Child postpartum use case there is no mention of a company. If such entity will appear, then this can be added to the custom data structure and process by the analyst. System can facilitate any process and user screen. .

3. Not to be considered	
developer to create using Python or JS/Node just regular code to do processing.	
The requirements mentioned here are quite specific. To cater to these requirements, a new type of no-code development platform may be needed and the existing no-code platforms may not suffice.	As mentioned in the specification there are multiple products selected as reference. According to GovStack methodology the requirements may be fulfilled with multiple products/BB-s. Can you specify where you found the limitation in question.
There are broadly two types of no-code platforms. No-code platforms that facilitate quick front-end development and no-code platforms that enable end-to-end application development. For the first category of no-code development platforms, the starting point is often APIs. They facilitate seamless integration with APIs by citizen developers (analysts) and they also allow rapid development of the front-end based on those APIs. For the second category of no-code development platforms, the entities along with their APIs (digital registries in our context) are created first, the APIs from those entities are used to build process flows and finally the front-end is also developed based on either the process flows or the APIs. It may be a better idea to check whether the requirements can be aligned with the way the existing no-code development platforms work.	Requirements are envisioned to fulfill the needs of a no-code platform (the second one). As mentioned in the document, the specification is considering an example found in UNCTAD system.

12 Future Consideration

12.1 Integration with a blockchain solution to guarantee the integrity of data and logs from unauthorized changes. This option may be available with a fee therefore should be optional.

12.2 Data storage federation (for example as a block chain)
Cloud storage federation model provides the integration of multiple cloud storage providers into a single virtual storage pool, eliminating the dependency on a single provider and decreasing vendor

lock-in problem. Moreover, federating multiple cloud storage providers improve data availability, storage scalability and data processing performance.

12.3 Installation automation requirements- system installation could be done with full automation in a local server or in a cloud.

12.4 Integrations with Electronic signature solutions to give a consent as an applicant or a confirmation as a registrar.

12.5 The registration information may be read out aloud and displayed on screen to the applicant as operator is entering registration data.

12.6 AI supported automated registrations and decisions.

12.7 Integration with chatbot and knowledge base to support users in their walkthrough in the service.

12.8 SMS and/or voice/natural language e-services as Plug and play BB components should be described.

12.9 Open data component to share all data that is anonymized. Publish appropriate (i.e. anonymised) Open Data automatically on registration applications, and regularly (e.g. monthly) pseudonymised aggregate data

12.10 Review results to be added to the next versions of the specifications. See decisions below:

Comments/feedback	**Suggested Action/Reason**
It seems, that specification currently focus only onto very simple registration cases and do not consider more complex processes, for example like financial services licensing.	Yes, the GovStack methodology is to focus on functionality needed for Use Cases assigned (Mother and Child registration program, Postpartum infant care). However, it is not hardcoded, therefore any process (including financial services) can be processed. The functional specification allows to create e-services involving multiple registrations and multiple entities from any domain.
However, Digital registries BB looks to me like DBMS specification. For example, PostgreSQL or ORACLE RDBMS are well fit to the Digital registries BB specification. My point is that if I will try to implement some registration process, for example, in Ukraine, where I recently were engaged, I do not see much added value from the set of two specifications.	Digital Registries is separated BB because not all Registration processes need a new database/registry. Registration BB and Digital Registries BB can operate as one or it can be separated into multiple functional BB-s. We decided to split the functionality in order to simplify the reading. If the Registrations BB is not suitable for your processes in a country in focus, then we could use your requirements for validating our BB specification. **Additional requirements can be added in the next iteration when new Use Cases are added.**

Comments/feedback	Suggested Action/Reason
The document is at a high level and doesn't make clear how a developer can take these specifications and implement.	This is meant to be a guideline and not a technical spec suchs as a SRS (Software Req Specification). Developers can choose to implement best dev. practices and ensure compliance with defined API-s and functionalities here. **With new Use Cases we can add more flesh to the skeleton.**
Should be a registration API, without needing to use the identified application	Additional API-s can be added based on next use cases. However, somebody needs to define the UI functions for the Registration BB. Without the UI the no-code principles do not work. **Technically the UI and API- in the back can be separated.**
In countries with existing interoperability platform there is an option that some documents, which are needed for registration can be obtained from other state agencies. For example, 15 y.o. child wants to go on competition with national team from Ukraine to Turkey. To cross a boarder with team coach the child should have authorisation from parent. In Ukraine in such situation parent go to Notary and Notary system can automatically lookup from Civil (Population) Registry the fact that the person is indeed a parent of a child. That means that the development platform should have a Transformer to build an adapter, which provides connectivity from the registration workflow to external state or even private sector organisation to lookup for evidences, which are needed for registration.	Yes, this is BOT function and is already described in requirement SF-7. Validation: one or more human or automated ("robot" or "BOT role") operators can review the claims provided, approve or reject an application, send claims to a registry and issue a credential (we call this part: "processing"). This is already described. **However, more requirements can be added in the next iteration.** Back office system registers all applications submitted. If configured so, the process flow may have bot(s) and human(s) in the flow. Bot roles process applications by validating information against external API information. When processed the BOT role will pass the processing task to the next role. **Requirements can be extended during next use cases.**
One subcomponent here should be a component to create requests to third parties and prepare adapters for receiving responses. That would support the Once-Only principle: a person should not submit to registrar information, which already exists in some other state agencies.	The function that you seek is part of UI and flow builder (**Mapping**). See BOT function described in requirement SF-7. We **will add more requirements** on this function in the next phase. It may be a separate module or even BB.
There are situations, when workflow registration is one organisation and the result of registration goes to a Registry, which is different organisation. For such cases there	Yes, there are two options for this: 1. Use Roles to process the application inside the same BB web user interface for external operators (SF-3).

Comments/feedback	**Suggested Action/Reason**
are should be capabilities to define external messaging.	2. Use Data bots to send data/validate to external endpoints (BB-s). Bot roles process applications by validating information against external API information. When processed the BOT role will pass the processing task to the next role. Although, some additional requirements may be needed to communicate with the **Messaging and Workflow BB.** This functionality can be described in the next iteration. Decision- will be added to the Future consideration chapter.
I do not see any requirements regarding layout of forms. For example, here is one Axure prototype for one very simple license: https://px8vs8.axshare.com (when you click on link, on top bar in left-upper corner there is is kinda hamburger icon - press it and left-side menu will appear to go through screens). This is simple license UI prototype, which has 29 pages in 4 sections: 2 sections for the self-service part and 2 parts for back-office processing. I doubt that I will be able to implement those screens using tool, which is only has compliance to those 8 requirements.	Please see requirement SF-1, SF-2 and SF-3. All the requirements explain the functionality how the User Interface will be built by the Analyst. In general the system must facilitate similar business requirements as envisioned in the Axure prototype. In this document we have envisioned at least four pages/forms (MVP): Guide, Applicant Form, Document upload, payment, Send. In the next iteration we can add functionality to add any number of pages/screens. Decision: will be added to this version- **Analyst can add any number of pages to the user screen flow.**
SF-7 Does that mean that there is also a tool to define an API request, including to map for fields to request fields and provide URL and authorisation details? Where this tool requirements are?	Yes, Bot action is about API request and **mapping** of fields. The Open API endpoint mapping must be solved by this BB in order to fulfill the requirement to pull data to the form and to send the data to external endpoints. SF-7 is the only requirement, The rest of the functionality to achieve this must be done by the BB developers. And as the communication is done via Information Mediator, (Pre-requirement) the authorization details are already configured for the Open API endpoints(See Information Mediator BB). **Will be added to the specification: Open API endpoint URL defining and API request/response mapping functionality.**
5.2 I would suggest to add also service-level management capabilities in the Operator organisation. Management should be able to establish quality requirements on how quickly processing should be done and should be able to monitor fulfilment of established service quality targets.	**Service level management** is a good idea, we will add it to the future iteration. We can add the objective and measure the performance.

Comments/feedback	Suggested Action/Reason
5.2 In complex cases, like for example licensing of financial services, registration process may be quite long process with a lot of interaction between Registrar and Applicant. For such processes there should be capability for bi-directional communication between parties in context of different fields of application.	**Asynchronous bi-directional communication between the applicant and operator** may be a good addition to the functionality. This will be added to the next iteration.
DS-6 in financial sector licensing process there are may be applications with many persons involved (like directors, MLRO, controllers etc.), who should provide its own input to one application. Such capability for multi-authoring of one application should be also presented.	In this first MVP iteration, there is no such business requirement in the Use Case. However, the **multi-authoring functionality for Part A users** may be excellent addition to the Registration BB. We will add it to the next iteration.
These specifications do not handle Inclusion issues - Multiple languages, assisted model, differently abled people, etc.	Language of the user interface should be cross-cutting requirement. Will be discussed where to add.
Specification on encryption for data storage - How will the business user specify which fields need to be encrypted?	Will be added in the V2. The business user/analyst can configure the captured data properties and decide the need for **encryption in rest** state.