This version of the template has been deprecated. Please see https://docs.google.com/document/d/1TRH-GuEMVKnSOFRL1bKlW5lV_b5VLjkh/edit#heading=h.4i7ojhp

Building Block Definition

<Name of Building Block>

Version 1.0.0

Developed by: <Names and organization affiliations of working group members>
In cooperation with GIZ, ITU, DIAL, and the Government of Estonia

1. Description:

<Example Description>
Registration services attribute a unique functional ID to a person, place or other entity to identify and access information about it. According to the World Bank, functional IDs are those that evolve out of a single use-case, such as voter IDs, health records, or bank cards, and are created with a specific purpose in mind, differing from foundational IDs which are created with a general purpose in mind. Registration services can also use the foundational ID or map it to the functional ID where such an identity exists. Examples of specific registration services include immunization, disease and citizenship records, as well as birth and death registration. The ensemble of utilities for capturing, recording, profiling, searching, retrieving and verifying this identity information is encapsulated as registration services. The information itself will be deposited into and retrieved from corresponding functional registries (see the Registries ICT Building Blocks). Registration services help profile entities by enabling the registration of different categories or groups and documenting their access to various services. These services also onboard users into a programme or service offered by an organization (eg rural advisory service), capturing related demography, profile and citizen ID information.

2. Key Digital Functionalities:

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

2.1. Examples for Identification Building Block

Foundational IDs come with no specified purpose or attached entitlement but functionalities simply let an entity prove who it is
Captures only limited information about users, such as name, date of birth, address and gender
For a given set of credentials, fetches a corresponding ID if it exists in the registry
Uses different biometric methods to identify and authenticate users through means other than user photographs (eg fingerprints, iris scans, facial recognition) to ensure there are no duplicates or fakes, creating a highly trustworthy database
This ICT Building Blocks also uses a publicly available interface, or open API, which allows any licensed service provider to verify if users are who they claim to be
Used to enable services such as opening bank accounts, buying SIM cards, receiving entitlements from the government, signing forms electronically, investing in mutual funds and getting credit
Incorporates privacy into its design when the purpose of the authentication is not revealed if a service provider sends an authentication request.

2.2. Examples for Registration Building Block

Assign a unique (functional) ID and create identification pieces (eg smart card, bar functionalities code, RFID, digital token, unique number) with that ID and all related information about the entity
Map the assigned new (functional) ID to existing national (foundational) ID, if any
Populate demographic and/or geographic data of entities when registering for the first time
The complete registration information will be further used for any transaction or services later
This information can be used to complete the know your customer (KYC) requirements that businesses use to assess potential clients and comply with regulations
Capture identification information from biometrics, photos, scanned images, typed input, etc, and update the relevant registry with these entries using a specific ID tag
Determine if the specified ID exists in the registry according to a match with the given credentials
Automatically identify possible duplicate entries and merge them
Retrieve information corresponding to queries for a chosen entry, such as eligibility for subsidy, enrollment status, profile
Retrieve information of entries in the registry matching or nearly matching given credentials

2.3. Examples for Data Collection Building Block

Uses different data collection devices such as mobile phones, tablets and sensors functionalities
Requires identification and authentication to ensure that the system considers only data captured by authentic sources
Customizes data collection forms to enforce data quality, integrity and business Workflows
Collects multimedia content (eg video, audio and photos) and records geolocation coordinates
Uses a data collection form or interface that can include access to a camera or a sensor to capture ambient information
Sets data definitions and standards such as measurement system, format and nomenclature (eg population census, definition of income groups, standards related to type of housing, and terminology such as literate and semi-literate)
Provides data analytics tools to create statistics derived from collected data
Collects data offline and synchronizes data with a central data repository once connectivity is established
Provides safeguards against data loss
Provides access to data through APIs

3. 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. This section will describe any additional cross-cutting requirements that apply to this building block.

Cross-cutting requirements will use the same language (MUST or SHOULD) as specified in the architecture document.

<Example Cross-cutting requirements>

3.1 Privacy

Personal data MUST be kept private and never shared with any parties, except where specific authorization has been granted

3.2 Audit Logging

Logs MUST be kept in a database of all records that are created, updated, or deleted. Logs MUST include timestamps and identify the user and affiliation that performed the transaction

3.3 Source Code

Source code SHOULD be available and easily accessible

3.4 Security requirements

MUST use 3FA for transactions accessible via mobile

Functional Requirements:

The functional requirements section lists the technical capabilities that this building block should have. These requirements should be sufficient to deliver all functionality that is listed in the Key Digital Functionalities section.

These functional requirements do not define specific APIs - they provide a list of information about functionality that must be implemented within the building block.

These requirements should be defined by subject-matter experts and don't have to be highly technical in this section..

<Example Functional Requirements>

Requirement	**Type (Must/Should/May)**
The registration system will allow users to create and update programs (also referred to as initiatives - a use case or project that requires registration of persons or facilities)	Must
The registration system will allow users to create and update registrations of entities (either persons or facilities) into a specific program	Must
User can query existing registrations based on search criteria. Users should be able to query a single record or multiple records	Must
The registration system will have a roles-based permission system to determine which users can perform which operations	Must
The registration system should have search and matching capabilities to avoid duplication of existing records The rules for matching records should be configurable	Should
The registration system can provide reports and listings of all entities registered in a specific program	May

Data Structures:

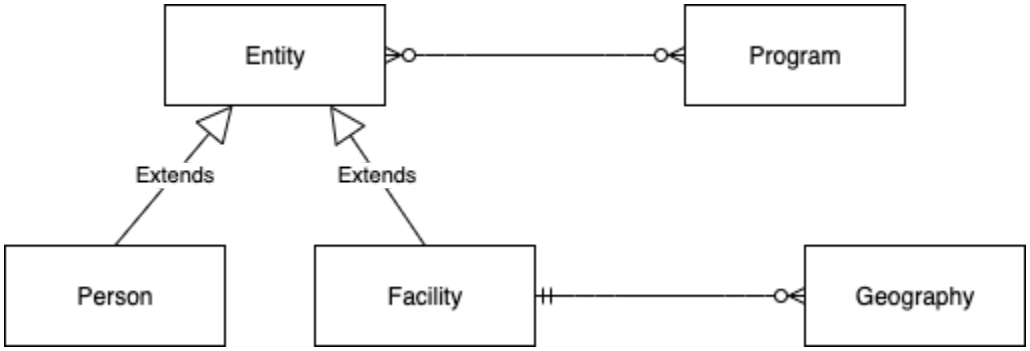
Resource Model:

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

Note: Recommend using https://app.diagrams.net/ to create the resource model and store in BuildingBlock repository (https://github.com/GovStackWorkingGroup/BuildingBlockAPI)

<Example Resource Model>
The registration building block's core data model is an Entity. Based on the use case, this entity may be defined either as a Person or a Facility. A registration record connects an Entity to a specific Program. Finally, a Facility/Person entity can be associated with a Geography. (*Note: this model may be extended to include additional entities, such as vehicles or equipment).*

Note that a Person or Facility entity can be further expanded based on the specific use case. For example, a Patient entity may be defined (derived from Person) for a health use case.



Standards:

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

<Example Standards>

All dates should follow ISO 8601.

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. Information about data elements will include:
Name
Description
Data Type
Required/Optional flag
Link to applicable standard(s)
Notes

Note: The data elements for each entity in the resource model will also be represented in the OpenAPI schema (link to GitHub repository)

<Example Data Elements>

Person

Name	Description	Type	Required	Standards	Notes
ID	Unique identifier	Integer	Y		Generated by building block on creation
Surname	Family name	String	Y		
First Name	First name	String	Y		
Birth Date	DOB	Date	N	ISO 8601	
Mobile Number	Phone number (mobile) of user	String	Y	E.164 Mobile number standard	Should include country code
Government ID number	Government issued ID number	Integer	N		Used when linking to global ID building block

<Example Model Schemas>

Each model schema MUST have a corresponding JSON Schema. The API definition file can be found here: https://raw.githubusercontent.com/GovStackWorkingGroup/BuildingBlockAPI/main/ExampleSchema.json

Changes to model schemas can be made by submitting a Pull Request on this repository.

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.

All APIs will be defined using the OpenAPI (Swagger) standard. The API definitions will be hosted outside of this document. This section may provide a brief description of required APIs. This section will primarily contain links to the GitHub repository for OpenAPI definition (yaml) files as well as to a website hosted by GovStack that provides a live API documentation portal.

<Example Service APIs>

Any implementation of the registration building block must support the APIs documented at this site: <link to GovStack API portal>

The API definition file can be found here: https://github.com/GovStackWorkingGroup/BuildingBlockAPI/blob/main/ExampleAPI.yaml

Changes to the API definitions can be made by submitting a Pull Request on this repository.

Workflows:

A workflow provides a detailed view of how this building block will interact with other building blocks to support common use cases.

This section lists workflows that this building block must support. Other workflows may be implemented in addition to those listed.

Standards:

The workflows MUST adhere to all standards defined in this document as well as in the GovStack architecture document (link to appropriate section in architecture document)

<Example workflow>
Name: Registration of new person

Description:

This workflow is used to enter a new record in the registration database. Data is submitted from a front-end application. This workflow shows a connection to an Identification building block (such as a government ID system) to validate that the registration is associated with a real person.

Interaction with Other Building Blocks:

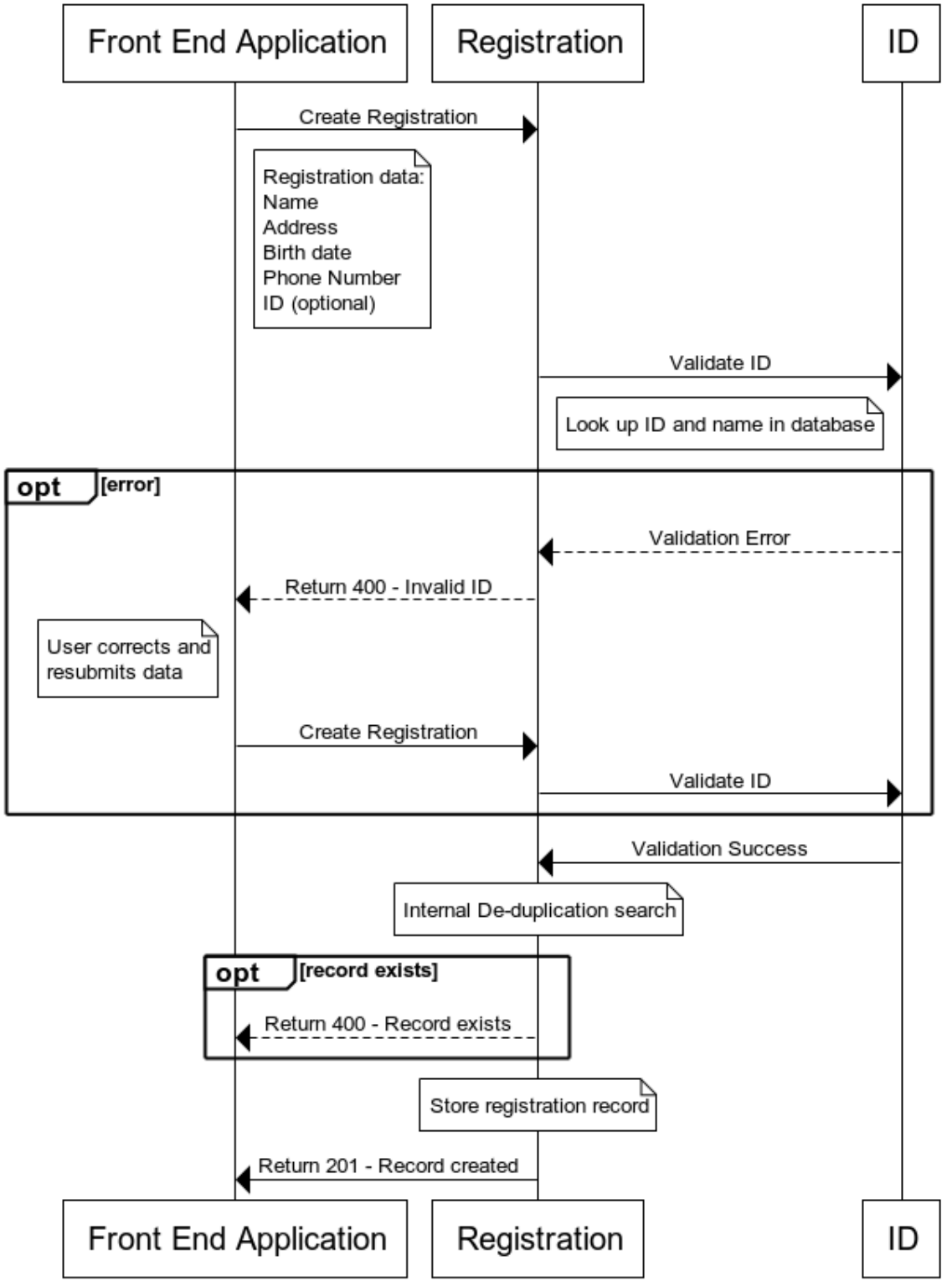
This workflow requires interaction with the *User Interface* and *Identification* Building Blocks

Sequence Diagram:

The sequence diagram shows the flow of data between building blocks for this workflow.

The diagram is created with a tool like https://www.websequencediagrams.com/ The raw text used to create the diagram should be saved and stored in the GitHub repository (https://github.com/GovStackWorkingGroup/BuildingBlockAPI).

Registration Workflow



www.websequencediagrams.com

Interactions

Name	Required Data	Notes
Create Registration	Person Data Structure	Data structure may contain additional elements
Validate ID	<Link to appropriate data structure in ID building block>	Inform user if record already exists and return unique identifier
Internal De-duplication Search	Person Data Structure	Return a 'Record Exists' message to the user if it is already in the registration database
Create Registration Record	Person Data Structure	Generate unique ID for this record (auto-increment)

<The next workflow would go here, including name, description, sequence diagram and interactions>

Other resources:

Link to architecture requirements document (and specific sections within that document, such as cross-functional requirements, general recommendations)

Link to use cases document – this document may be a valuable resource while developing workflows to ensure that a variety of different use cases are covered by the building block definition.

Link to BB criteria and maturity metrics document created by Tanvir

Link to Low Resource Settings document

Link to GitHub repository and OpenAPI documentation site for the building blocks.