DATA TRIANGULATION

Functional Requirements Specifications

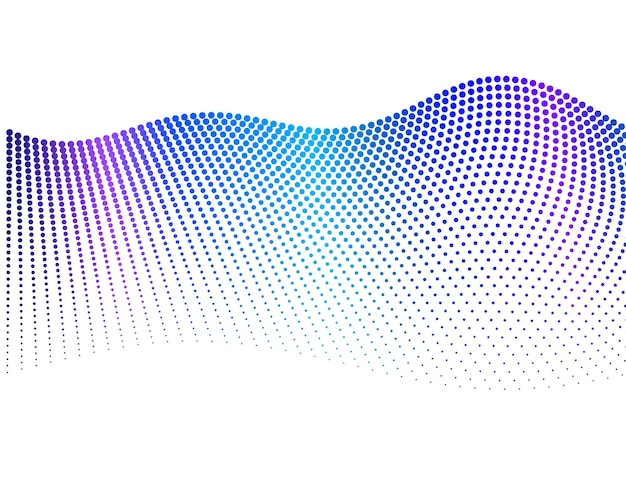


TABLE OF CONTENTS

[**INTRODUCTION** 3](#_Toc163560514)

[**PURPOSE** 3](#_Toc163560515)

[**OBJECTIVES** 3](#_Toc163560516)

[**DATA TRIANGULATION FUNCTIONAL REQUIREMENTS SPECIFICATION** 3](#_Toc163560517)

[**STEP 1: INGESTION** 3](#_Toc163560518)

[**STEP 2: PRE-PROCESSING** 4](#_Toc163560519)

[**STEP 3: MULTI DATASET MATCHING** 8](#_Toc163560520)

[**STEP 4: POST PROCESS** 13](#_Toc163560521)

[**STEP 5: OUTPUT** 13](#_Toc163560522)

[**DATA FLOW AND BUCKETING STRUCTURE** 13](#_Toc163560523)

[**MICROSERVICE CONSTRUCT** 15](#_Toc163560524)

# **INTRODUCTION**

This document is a functional requirement specification for Data Triangulation Algorithm. The main purpose of this algorithm is for the data quality and data validation for the beneficiary identification in the schemes or any benefits of schemes being delivered by the government on a single platform. The purpose for this is to validate the data and data quality for the beneficiary identification.

# **PURPOSE**

This document aims to capture the functionalities of Data Triangulation to enable beneficiary identification in a comprehensive system.

# **OBJECTIVES**

The main objective of data triangulation is to conceptualize and develop an algorithm on how name-pairs may be matched using fuzzy matching to obtain potential matches in the base dataset (golden record dataset) and any reference dataset.

# **DATA TRIANGULATION FUNCTIONAL REQUIREMENTS SPECIFICATION**

The overall flow of the data triangulation is envisaged as below:

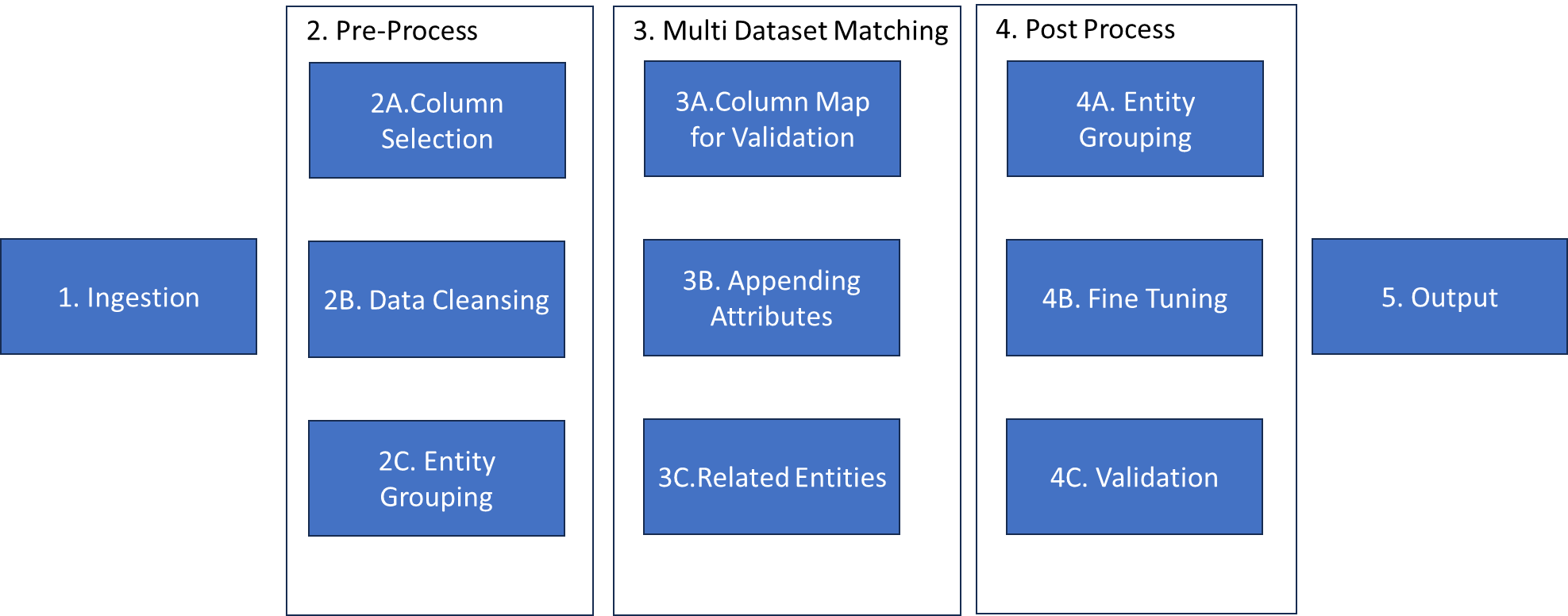


Figure 1: Overall Flow of Data Triangulation

The above entire flow is further sub divided into various modules as below:

## **STEP 1: INGESTION**

This phase involves the intake of data from various data sources (Base Data Set/Reference Data Set) into the system.

* 1. Data shall be ingested from different databases, files or other sources of data.
  2. The system should allow merging and splitting of records within a dataset or across multiple datasets at this step.
  3. The system should identify and select the recent record of citizen data from the dataset(s) (if available).
  4. The system should support different ingestion methods and formats (.xls, .xlsx, .xml, .csv, JSON, etc.) to accommodate diverse data sources.
  5. The system should be able to support large volumes of data in unstructured and structured formats.

This system should be designed such that this step to be run in iterative manner with any number of iterations, with or within dataset(s).

**Output expected at this step:**

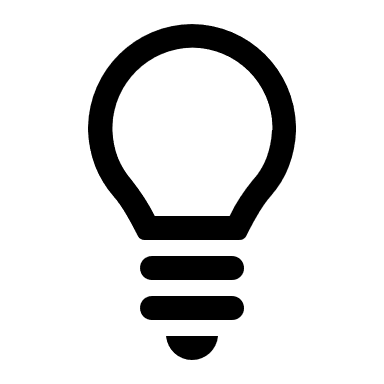
The system must return a profile summary of the ingested data vs. the output data for both the base and reference datasets after processing at this step. This includes a report in a tabular format as indicated below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Input** | | **Output** | |
| **Field** | **Volume** | **Field** | **Volume** |
| **<<Base Dataset>>** | Field 1 |  | Field 1 |  |
| Field 2 |  | Field 2 |  |
| ….. |  | ….. |  |
| Field n |  | Field n |  |
| **<<Ref. Dataset>>** | Field 1 |  | Field 1 |  |
| Field 2 |  | Field 2 |  |
| ….. |  | ….. |  |
| Field n |  | Field n |  |

The fields processed at each step have to be captured in the above table.

## **STEP 2: PRE-PROCESSING**

In this phase, the ingested data shall be selected, processed, cleansed and grouped using the following methods:

* 1. ***Column Selection:*** Users should be able to select specific columns from the ingested datasets based on their requirements. This allows for focusing on relevant data and reducing processing overhead.
     1. The system is expected to allow rule-based selection of columns or;
     2. ****The system should allow algorithm (AI/ML) based column selection.

**Use Case: Extract attributes to generated key for matching -**

Extract name along with attributes such as father name, mother name, gender, village (name and code), block (name and code), district (name and code), mobile number and Aadhaar number. The exact list of attributes shall depend on the fields present in the candidate database which is to be matched with the master database.

* 1. ***Data Cleansing:*** This involves identifying and correcting errors or inconsistencies in the data. The System should provide tools for data cleaning such as -
     1. Removing Duplicates entries: Identify and remove duplicate records within the dataset to ensure data integrity and accuracy. The system must be able to provide the degree of duplication to which the entries are found duplicate in the datasets.
     2. Handling Garbage values: Identify and handle irrelevant and non-logical data entries that do not contribute to the analysis and insights. This activity shall involve filtering out outliers or irrelevant data entries.
     3. Handling Missing Values: The system should implement strategies to handle missing or null values, such as missing values in records of citizens etc. or removal of records with missing values.
     4. Standardizing Formats: The system must ensure consistency in data formats across different attributes and/or datasets. This may involve converting the raw data into a common, standardized and consumable format (e.g. date (DD/MM/YYYY), numbers, gender, first name, suffix, last name, etc.) to facilitate easier comparison, merging and analysis.

e.g., ***Name Standardizing:*** Names are standardized based on similar sounding letters/patterns used in Indian names.

1. Replace adjacent similar characters by single character. E.g. “aa” replaced by “a”, “”bb” replaced by “b” etc.
2. Replace unigrams: Characters in names shall be replaced as follows:

|  |  |
| --- | --- |
| e 🡪 i | j 🡪 z |
| v 🡪 w | q 🡪 k |

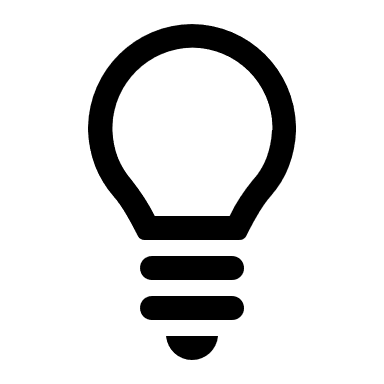
1. Replace bigrams:

|  |  |
| --- | --- |
| ph 🡪 f | gh 🡪 g |
| th 🡪t | kh 🡪 k |
| dh 🡪 d | ch 🡪 c |
| sh 🡪 s | au 🡪 o |
| bh 🡪 b | ks 🡪 x |
| ck 🡪 k | ah 🡪 h |

1. Remove ‘Consonant + a’ combination:

Remove the character ‘a’ from the name in case the same is preceded by a consonant.

|  |  |
| --- | --- |
| “PARDIP” or “PRADIP” 🡪 | “PRDIP” |

* + 1. Correcting data Inaccuracies: Identify and correct errors or inaccuracies in the citizen profile data, such as typo errors, misspellings, or incorrect values to improve data quality.
    2. Updating Recency: Ensure that the data reflects the most recent information available by updating event records with the latest citizen profile data. This may involve integrating real-time data sources or periodic updates between the datasets such as Aadhaar, PAN, etc.
    3. ****Data Parsing: The system must allow for parsing the complex data entries into separate fields and should also allow for converting the data into different data formats. E.g., Converting address field of citizen into multiple separate fields.

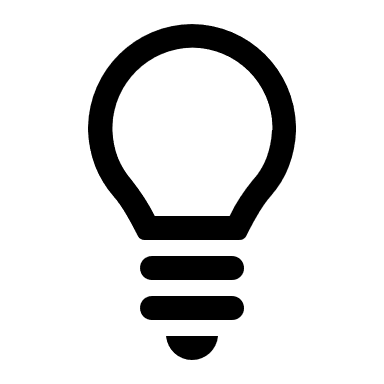
**Use Case:** **Clean all name attribute fields** including name, father name, mother name, village name, block name and district name in base and reference datasets, including:

1. Making all the names in upper-case.

|  |  |  |
| --- | --- | --- |
| **“Sulochana”** | **🡪** | **“SULOCHANA”** |

1. Remove salutations from the names as per the patterns specified below:
   * “Mr.<space>” and “Mr<space>"
   * “Mrs.<space>” and “Mrs<space>”
   * “Ms.<space>" and “Ms<space>”
   * “Miss.<space>" and “Miss<space>”
   * “Master.<space>" and “Master<space>”
   * “Dr.<space>" and “Dr<space>"
   * “Smt.<space>" and “Smt<space>”
   * “Sh.<space>" and “Sh<space>”
2. Remove relationship markers from the names:

* “S/O”, “SO<space>”,”S O”
* “D/O”, “DO<space>”,”D O”
* “W/O”, “WO<space>”, ”W O”

1. ****Remove numbers and special characters (excluding <space>) from the name strings.

**Use Case: Remove Stop words** **–**

Names may contain certain region-specific suffixes. The following suffixes shall be removed:

* “devi”, “dei”,”debi”
* “kumar”,”kumaar”,”kumari”,”kumaari”,”kmr”,”kumr”
* “bhai”,”bhau”,”bai”
* “ben”
* “singh”,”kaur”
* “Md”, “Mohd”, “Mohammad”, ”Mohamad”

The system is expected to run the above processes for all fields such as mobile number, name, last name, Government issued identity numbers, Date of birth, age, email id, income, gender, etc. to ensure adherence to standard formats and a logical citizen profile is created.

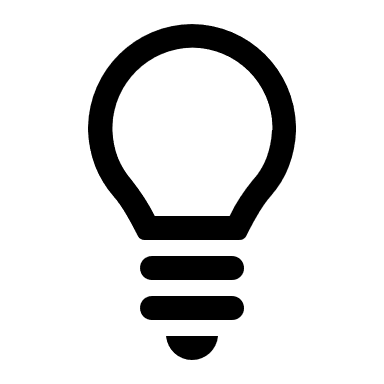
**Output expected at this step:**

The system must return a summary of the ingested data vs. cleansed data by employing above techniques for both the base and reference datasets after processing at this step. This includes a report in a tabular format as indicated below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Input** | | **Output** | |
| **<<Technique>>** | **Volume** | **<<Processed o/p data>>** | **Volume** |
| **<<Base Dataset>>** | Field 1 |  | Field 1 |  |
| Field 2 |  | Field 2 |  |
| ….. |  | ….. |  |
| Field n |  | Field n |  |
| **<<Ref. Dataset>>** | Field 1 |  | Field 1 |  |
| Field 2 |  | Field 2 |  |
| ….. |  | ….. |  |
| Field n |  | Field n |  |

The fields processed at each step have to be captured in the above table.

* 1. ***Entity Grouping:*** The system should be capable of grouping the common attributes or identifiers in the citizen profile data based on custom rules or AI/ML based logic. The grouping should provide trends and analysis from the activity to assist in data driven decision making.

** Use Case:** **Generate Candidate pairs**

Matching every name in the source database with each name record in the reference will not only be computationally extensive and time consuming but will also create many name pairs. Hence names to be matched shall be filtered and candidate pairs shall be identified for matching based on the following criteria:

1. Unique key:

Identify a field(s) which have a unique value for each record in the respective databases. Incase both the databases have same unique identifiers (such as Aadhar number, voter id number etc.) available, then 1:1 candidate pairs shall be generated and matching shall be done across the identified record only.

1. Geographical indicators:

Incase similar unique identifiers could not be found across datasets, then records to be matched shall be filtered based on the village/block/MC. Hence, names shall only be matched in case they belong to the same village/block/MC according the two datasets.

Using a phonetic fuzzy match algorithm ‘Metaphone’, on a combination of parameters including name, father name, mother name and spouse name and a threshold value of 50, generate a maximum of 3 candidate pairs. In case more than 3 candidate pairs are created, then select the top 3 based on the matching percentage.

The system should be designed so as to allow pre-processing step to be iterative in nature with provision of allowing any combination of processing within or with dataset(s).

## **STEP 3: MULTI DATASET MATCHING**

The next step in triangulation of data involves matching the processed data from the step 2 between reference and base datasets through the following:

* 1. ***Column Map for Validation:*** The system must allow users to define rules for mapping between columns from multiple reference datasets to perform validation process. The system to ensure user be able to identify source column(s) mapped to the destination columns and allow for matching the data across reference datasets.

Once the data is identified and mapped, each data should be assigned a weightage/ tolerance level/ confidence level to ascertain the authenticity of the data.

The candidate pair with the highest weighted average score (in case of multiple candidate pairs) shall be considered as the matched record for the corresponding record present in the base dataset.

Once the data is mapped to the destination columns and weightage assigned, the system must provide the user with options of operators for mapping such as:

* + 1. ***Like matching***: The system should allow comparing values using pattern-matching techniques to identify patterns or substrings within citizen data entries. This technique is required to match values with slight variations or discrepancies in formatting or structure. E.g., partial matches based on wildcard characters such as
       - ***'%'*** [matches any sequence of characters] *or*
       - ***'\_'* [**matches any single character]

The system must be able to employ the following algorithms –

* Metaphone *(‘Metaphone’)*
* Ratcliff/Obershelp Pattern Recognition algorithm *(‘SequenceMatcher’)*

**Output expected at this step:**

The system must return a summary of the input and output data fields which was identified and matched using the like-matching algorithm for both the base and reference datasets. The output should be visualized as below:

|  | **Input** | | **Output** | |
| --- | --- | --- | --- | --- |
| **Fields** | **Volume of unmatched data** | **Fields** | **Volume of matched data** |
| **<<Base Dataset>>** | Field 1 |  | Field 1 |  |
| Field 2 |  | Field 2 |  |
| ….. |  | ….. |  |
| Field n |  | Field n |  |
| **<<Ref. Dataset>>** | Field 1 |  | Field 1 |  |
| Field 2 |  | Field 2 |  |
| ….. |  | ….. |  |
| Field n |  | Field n |  |

* + 1. ***Fuzzy logic-based matching:*** The system should enable fuzzy logic type of matching approach, which is more flexible that considers the similarity between values rather than requiring exact matches employing fuzzy matching algorithm to compare strings or numerical values based on their similarity, taking into account factors such as -
       - spelling variations,
       - typos,
       - synonyms or other forms of linguistic or numerical variation.

Fuzzy logic-based matching is useful when dealing with data that may contain discrepancies or variations that could affect direct matching. The system must be able to employ the following algorithms –

* Levenshtein algorithm *(‘FuzzyWuzzy’)*
* Jaccard Algorithm *(similarity matching algorithm)*

**Output expected at this step:**

The system must return a summary of the input and output data fields which was identified and matched using the Fuzzy logic-based matching for both the base and reference datasets. The output should be visualized as below:

|  | **Input** | | **Output** | |
| --- | --- | --- | --- | --- |
| **Fields** | **Volume of unmatched data** | **Fields** | **Volume of matched data** |
| **<<Base Dataset>>** | Field 1 |  | Field 1 |  |
| Field 2 |  | Field 2 |  |
| ….. |  | ….. |  |
| Field n |  | Field n |  |
| **<<Ref. Dataset>>** | Field 1 |  | Field 1 |  |
| Field 2 |  | Field 2 |  |
| ….. |  | ….. |  |
| Field n |  | Field n |  |

* + 1. ***Exact (equals to) matching:*** This system should be able to handle direct comparing of values in corresponding columns across datasets to check for exact matches. This method is straightforward and is expected to match corresponding column data precisely without any variations.

**Output expected at this step:**

The system must return a summary of the input and output data fields which was identified and matched using the Exact (equals to) logic-based matching for both the base and reference datasets. The output should be visualized as below:

|  | **Input** | | **Output** | |
| --- | --- | --- | --- | --- |
| **Fields** | **Volume of unmatched data** | **Fields** | **Volume of matched data** |
| **<<Base Dataset>>** | Field 1 |  | Field 1 |  |
| Field 2 |  | Field 2 |  |
| ….. |  | ….. |  |
| Field n |  | Field n |  |
| **<<Ref. Dataset>>** | Field 1 |  | Field 1 |  |
| Field 2 |  | Field 2 |  |
| ….. |  | ….. |  |
| Field n |  | Field n |  |

The data summary of the output from the above matching techniques, which ever is utilized for the specific requirement has to be shown by the system to the user.

Through the above techniques, match the individual fields present in the base dataset with the corresponding fields of the reference dataset for the identified candidate pairs.

* 1. ***Appending Attributes:*** Once the data is matched between the corresponding columns using the techniques enlisted above, now the system must ensure the completeness of the attributes present or missing in the citizen profile data making the citizen/candidate data rich with all the fields present.

The system must be able to identify the missing attributes in the datasets by employing a combination of process of checking, validating and appending information across datasets matched to create a single, data-rich profile of citizen to ensure the standardization in format is maintained.

An Identity Resolution Process is to be employed in appending of attributes should consist of the following steps:

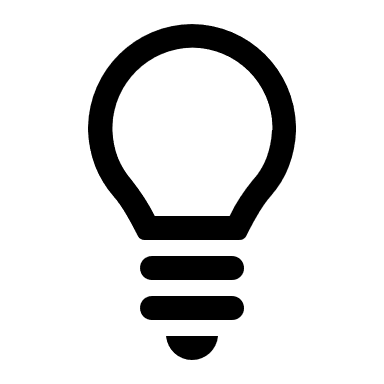
* + - 1. *Identification of data entities:* Identify the columns and attributes in the dataset
      2. *Connect:* identify the correlation between the different columns in the reference and base datasets.
      3. *Match:* Based on a defined set of attributes, create a data linkage between the records of the same person from several datasets or within one dataset.
      4. *Validate:* Using the datasets, validate the identity of the citizen profile and create sense out of it.
      5. *Activate:* create a single, data rich citizen profile.

**Output expected at this step:**

The system must return a summary of the input and output data fields which had missing attributes, and the updated/added attributes post processing at this step for both the base and reference datasets. The output should be visualized as below:

|  | **Input** | | **Output** | |
| --- | --- | --- | --- | --- |
| **Fields** | **Volume of missing <<attributes>>** | **Fields** | **Volume of appended <<attributes>>** |
| **<<Base Dataset>>** | Field 1 |  | Field 1 |  |
| Field 2 |  | Field 2 |  |
| ….. |  | ….. |  |
| Field n |  | Field n |  |
| **<<Ref. Dataset>>** | Field 1 |  | Field 1 |  |
| Field 2 |  | Field 2 |  |
| ….. |  | ….. |  |
| Field n |  | Field n |  |

* 1. **Related Entities:** From the above process, the system should be able to identify and establish relationships between matched entities across multiple datasets. During this process, the linked data records to be enhanced using the relevant available information from the third-party or internal datasets by parsing, standardization and record linking.

** Use Case: Name Match**

The name fields (name, father name, mother name and spouse name) shall be matched by generating similarity scores from the algorithms mentioned above. Based on the values received from each of the four algorithms, the match score for the name field shall be determined.

The system should be designed to allow multi-dataset matching in iterative manner and provide functionality of matching data/records in any fashion within or with dataset(s).

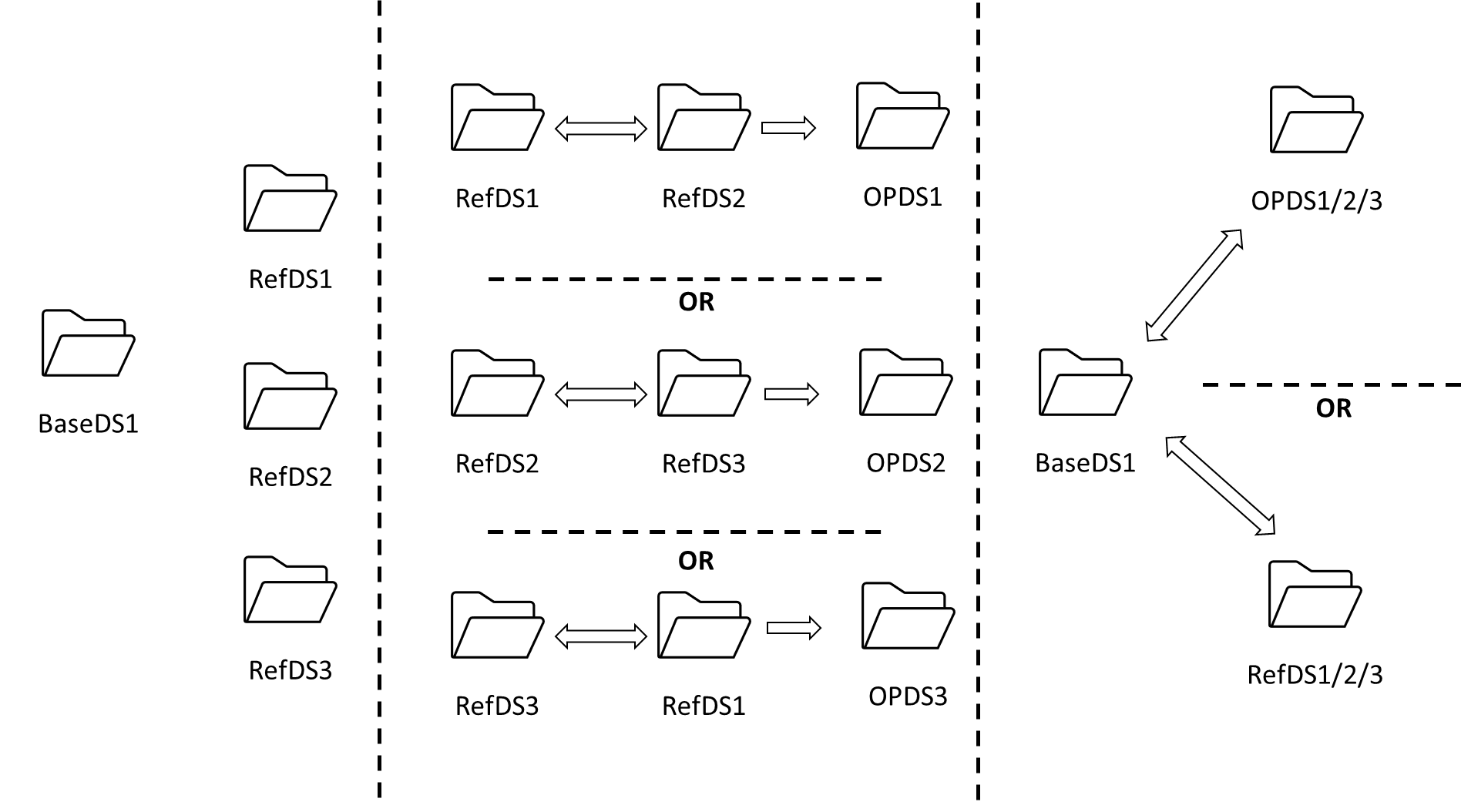


Figure 2: Possible combinations for Multi-dataset matching

## **STEP 4: POST PROCESS**

This process involves further processing of the citizen records by employing procedures such as grouping of the entities, fine tuning the grouped entities and validation for accuracy of the consumable records of data.

* 1. **Entity Grouping:** From the processed data of step 3, further refinement of entity grouping may be required based on the results of the multi datasets matching phase. Here, the system must group the entities based on common factors.
  2. **Fine Tuning:** Fine Tuning involves optimizing parameters or algorithms to improve the accuracy and efficiency of the data triangulation process.
  3. **Validation:** Validate the results obtained from the data triangulation process to ensure accuracy and reliability. This may involve comparing the output against known standards or conducting further analysis.

The system should be designed to accommodate processing multiple processed datasets in various combinations to allow for customized business requirements.

## **STEP 5: OUTPUT**

The system should provide the final name matched fields enriched with data field attributes (name, father name, mother name and spouse name) and be able to present the data in various output options such as reports, visualizations, or exportable datasets (supporting various formats such as. xlx, .xlsx, .xml, JSON, .csv, etc.). The output should also return the match score values returned by each of the four algorithms and store in a separate file along with matched metadata. The output should be customizable to meet different user requirements and should convey insights derived from the data triangulation process effectively.

# **DATA FLOW AND BUCKETING STRUCTURE**

The designed system is expected to follow the below data flow architecture for data ingestion at each step and the output from each step.

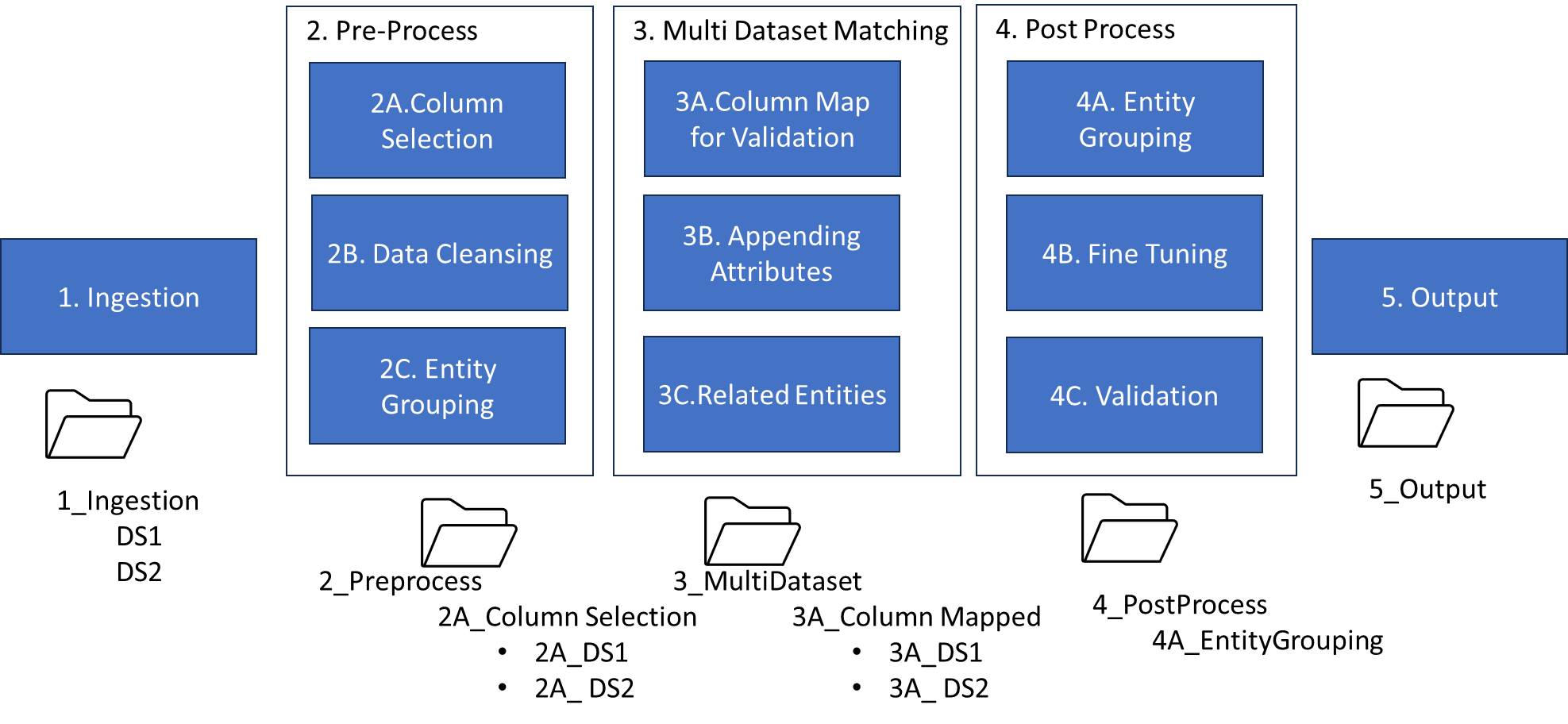


Figure 3: Data Flow and Bucketing Structure

The data ingested and processed at each step and the output from each step should be stored in folders as illustrated in figure 2. It is expected that the system must ensure naming convention that clearly depicts the input and output form each step both for base dataset and reference dataset(s).

Following the above data flow, the processing of data at each step is expected to follow the below structure for each of the base dataset and reference dataset(s):

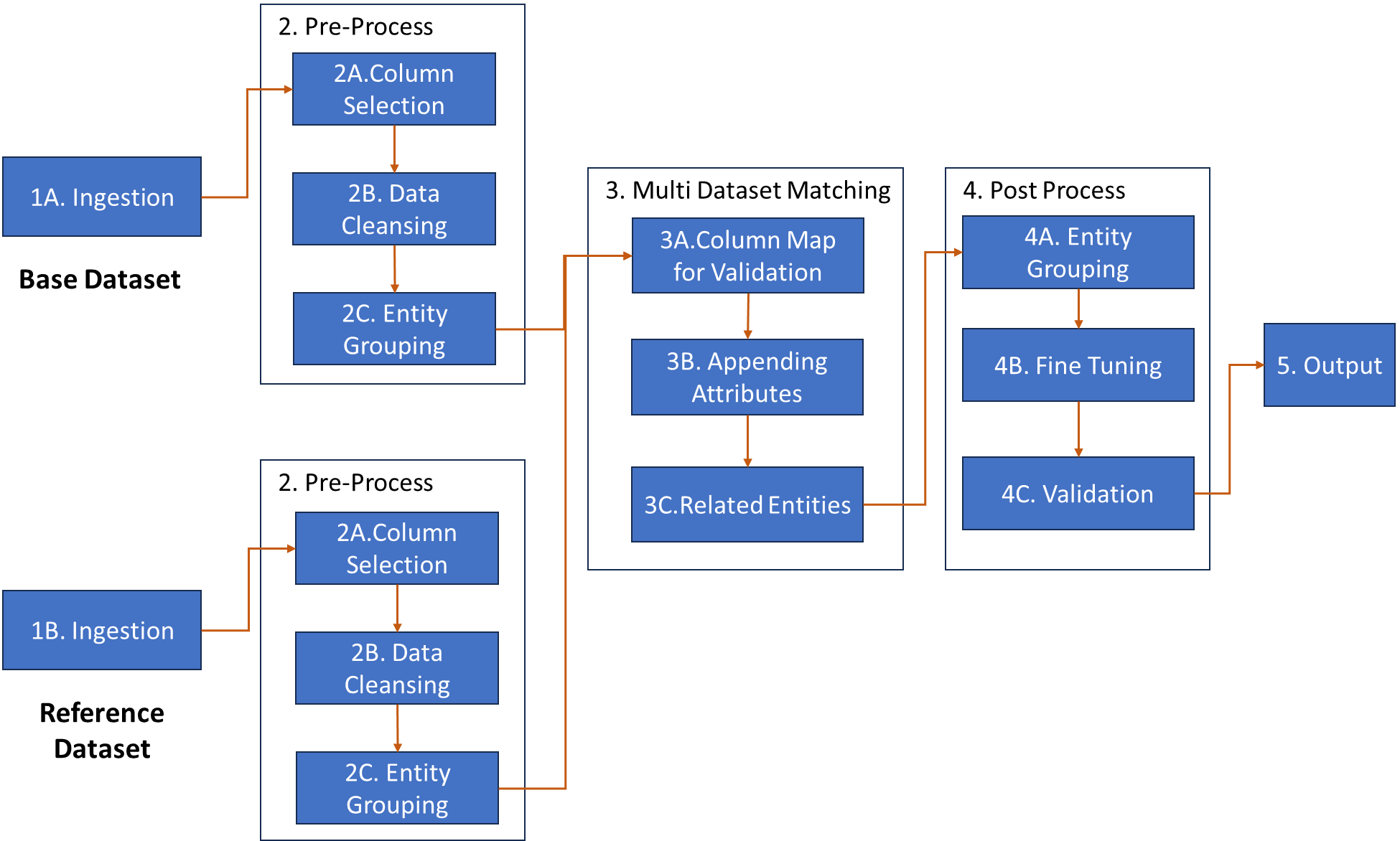


Figure 4: Data Flow at each step of Process.

# **MICROSERVICE CONSTRUCT**

The data triangulation system is expected to be designed and built using microservice architecture using APIs for each of the functional requirements enlisted above. The system should be capable of integrating all the microservices and create buckets(folders) and provide output of the process in visualizations or summary tables as required.

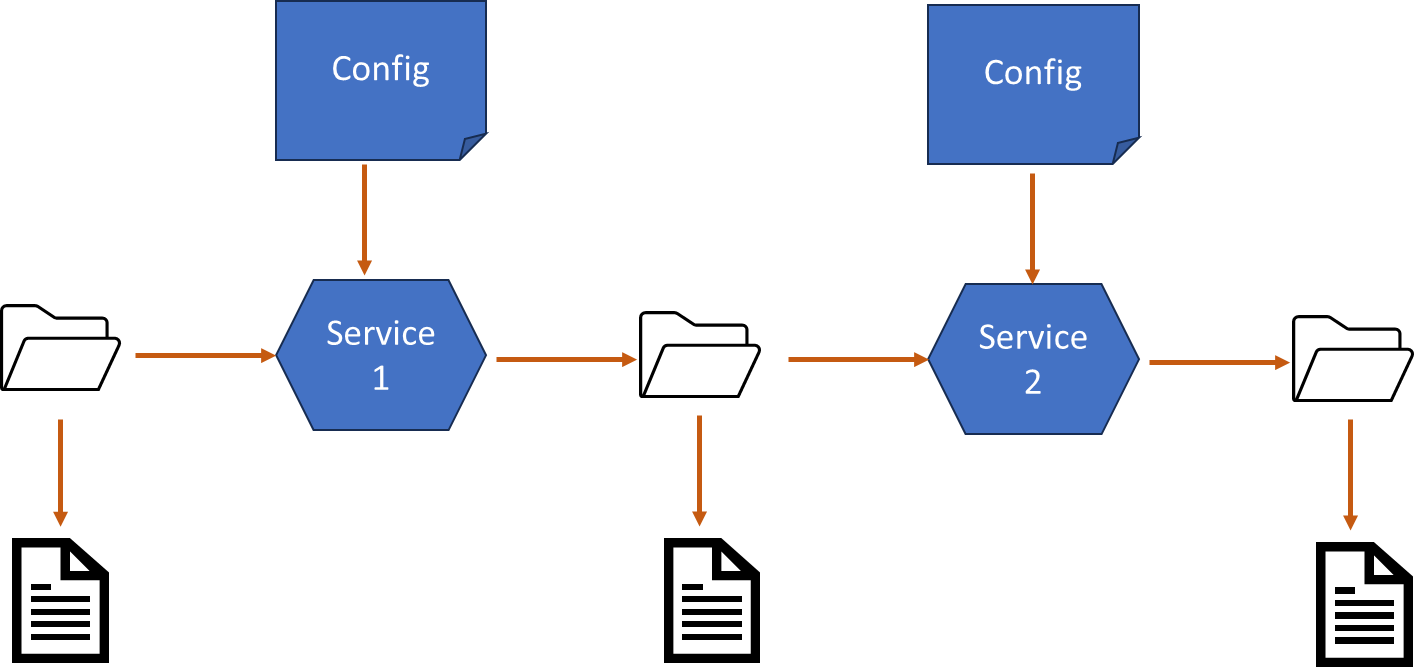


Figure 5: Microservice construct for designed system.

**My Understanding of Use Case**

This is a detailed functional requirements specification document for a Data Triangulation Algorithm. I'll break down the document into its main sections and provide a summary of each section.

**Introduction**

* The document outlines the functional requirements for a Data Triangulation Algorithm.
* The algorithm aims to validate data and ensure data quality for beneficiary identification in government schemes.

**Purpose and Objectives**

* The purpose is to capture the functionalities of Data Triangulation to enable beneficiary identification.
* The main objective is to develop an algorithm that matches name-pairs using fuzzy matching to obtain potential matches in the base dataset (golden record dataset) and any reference dataset.

**Data Triangulation Functional Requirements Specification**

This section outlines the overall flow of the data triangulation process, which is divided into five steps:

**STEP 1: INGESTION**

* Ingest data from various sources into the system.
* Support different ingestion methods and formats.
* Identify and select the recent record of citizen data from the dataset(s).
* Output: Profile summary of ingested data vs. output data for both base and reference datasets.

**STEP 2: PRE-PROCESSING**

* Select, process, cleanse, and group data using various methods, including:
  + Column selection
  + Data cleansing (e.g., removing duplicates, handling garbage values, standardizing formats)
  + Entity grouping
* Output: Summary of ingested data vs. cleansed data for both base and reference datasets.

**STEP 3: MULTI DATASET MATCHING**

* Match processed data from Step 2 between reference and base datasets using various techniques, including:
  + Column map for validation
  + Like matching
  + Fuzzy logic-based matching
  + Exact matching
* Output: Summary of input and output data fields that were identified and matched using each matching technique.

**STEP 4: POST PROCESS**

* Further process citizen records by:
  + Grouping entities
  + Fine-tuning grouped entities
  + Validating for accuracy
* Output: Refined entity grouping and validated results.

**STEP 5: OUTPUT**

* Provide final matched fields enriched with data field attributes.
* Offer various output options, including reports, visualizations, and exportable datasets.
* Output should be customizable to meet different user requirements.

**Data Flow and Bucketing Structure**

* The system should follow a specific data flow architecture for data ingestion at each step.
* Output from each step should be stored in folders with a clear naming convention.

**Microservice Construct**

* The data triangulation system should be designed and built using microservice architecture.
* Each functional requirement should be integrated using APIs.
* The system should create buckets (folders) and provide output in visualizations or summary tables as required.

Overall, this document outlines a comprehensive data triangulation process that involves data ingestion, pre-processing, multi-dataset matching, post-processing, and output. The system should be designed using microservice architecture and provide customizable output options.

**What is the Base data and what is the Reference dataset?**

The Base Data is the primary dataset, and the Reference Data are secondary datasets that are used to validate and enrich the base data through the data triangulation algorithm.

**The Base Data may contain attributes such as:**

* Name
* Father’s name
* Mother’s name
* Date of birth
* Address
* Aadhaar Card or PAN Card number

**Reference data may come from various sources such as:**

* Government Agencies
* Financial Institutions
* Healthcare Providers

**The Reference Data may contain attributes that are similar to or overlap with those in the Base Data, such as:**

* Alternative names or aliases
* Additional addresses or contact information
* Employment or income details
* Education or qualification records
* Other relevant information that can be used for validation or enrichment

The purpose of the Data Triangulation Algorithm is to match and validate the records between the Base Data and one or more Reference Data sets, using various techniques such as fuzzy matching, exact matching, and data cleansing. By doing so, the algorithm aims to:

**Identify duplicate or inconsistent records across datasets**

* Enrich the Base Data with additional information from the Reference Data
* Validate the accuracy of the records in both datasets
* Provide a more comprehensive and accurate view of the citizens, beneficiaries, or individuals