Moja Reporting Tool

DEVELOPMENT GUIDE

02 December 2020



# 

# Introduction

This document is an entry level guide into the architectural disposition of the Reporting Tool. Its role is to show the building blocks of the Reporting Tool and the rationale behind it. More technical explanations of the source code should be found within each Reporting Tool’s repository in Github.

This document is strongly related to a series of guides detailing how to set up and operate the Reporting Tool. We recommend that you have a look at them as well for a more fuller understanding of the domain.

[01. Reporting Tool - Environmental Set Up Guide](https://drive.google.com/open?id=1W3pb893CwDfrJxxLCxULwlmP5dBau-CpAamwiPFfluo)

[02. Reporting Tool - Build Guide](https://docs.google.com/document/u/0/d/1CU2ugodtOdO_JrVfVTx8OSE-sfUyg_f1SEAeDW-22nQ/edit)

[03. Reporting Tool - Installation Guide](https://docs.google.com/document/u/0/d/1xvTpzxOyuHvqk8YZN7-z0hm3infCWyBKHQxIlwsW1S8/edit)

[04. Reporting Tool - Operation Guide](https://drive.google.com/open?id=15njwuPlr-XH7zSASOeJVj0NcuadAwgoaY5_N0UKX-ls)

[05. Reporting Tool - User Guide](https://docs.google.com/document/u/0/d/1mvmMQbVS5ysvyNvdwvAqAI9K-NuQxW4AS2ieJPLWG7M/edit)

# Prerequisites

We need to meet two main preconditions in order to successfully contribute Reporting Tool’s development:

|  | Prerequirement | |
| --- | --- | --- |
| 1 | We need to have sufficient Java development skills | |
| 2. | We need to have sufficient Angular development skills | |

## 

# Outline

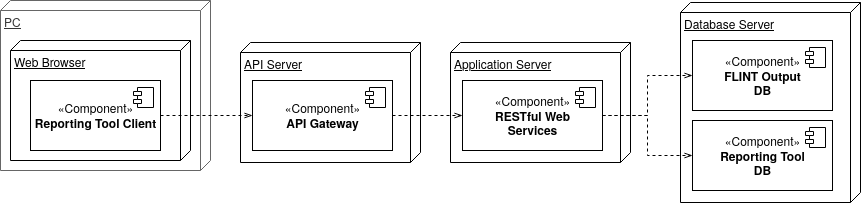
There are six things that we need to understand to contribute to the Reporting Tool’s development:

|  | Issue | |
| --- | --- | --- |
| 1. | The major design elements that make up the Moja Reporting Tool | |
| 2. | The functional decomposition of the above elements | |
| 3. | The execution elements of the functionally decomposed elements | |
| 4. | The tool’s source code organization | |
| 5. | The long term vision and direction for continuous improvement | |
| 6. | The logical next areas for improvement | |

We will explore each of the above concepts in detail in the following chapters.

# Major Design Elements

The Reporting Tool consists of four nodes hosting five major design elements. These are as outlined below with subsequent chapters giving more detailed explanations of each of the components.



*Figure 1: Reporting Tool’s Components*

|  | **Component** | | **Description** |
| --- | --- | --- | --- |
| 1. | Database Server Node | | |
| 1.1. | FLINT Output Database | | The FLINT Output Database in the first major design element of the Reporting Tool. It sits on the Database Server Node and is implemented as a PostgreSQL Database. |
| 1.2. | Reporting Tool Database | | The Reporting Tool Database in the second major design element of the Reporting Tool. It sits on the Database Server Node and is implemented as an Apache Cassandra Database. |
| 2. | Application Server Node | | |
| 2.1. | RESTful Web Services | | The Reporting Tool’s RESTful Web Services are the third major design element of the Reporting Tool. They sit on the Application Server Node and are implemented as Reactive Spring Webflux Microservices. They are packaged as Docker Containers which are in turn orchestrated by Kubernetes. |
| 3. | API Server Node | | |
| 3.1. | API Gateway | | The Reporting Tool API Gateway is the fourth major design element of the Reporting Tool. It sits on the API Server Node and is implemented using Traefik HTTP reverse proxy and load balancer. |
| 4. | Personal Computer Node | | |
| 4.1. | Reporting Tool Client | | The Reporting Tool Client is the fifth major design element of the Reporting Tool. It sits on Clients PCs as a Browser Based Application and is implemented as Angular Application |

## 

# Functional Decomposition

## FLINT Output Databases

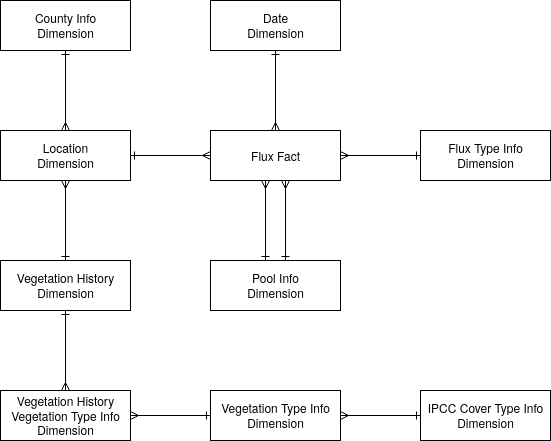
The FLINT produces three output databases that are available for use in the Reporting Tool: A Metadata, a FLUX and a STOCK database.

|  | Database | | Description |
| --- | --- | --- | --- |
| 1. | Metadata Database | | The metadata database contains auxiliary and summary information about the simulation. For example it includes:- Year Range, Spatial Range, Input Data/Databases Used, Modules and Configurations Used, Area/Count Of Simulation and finally Error Summary - Summary of the number/nature of errors produced during the simulation |
| 2. | Stock Database | | The stock database captures information on the amount of material in pools (e.g. carbon pools) |
| 3. | Flux Database | | The flux database captures information on the flux of material (e.g. carbon, nitrogen) between pools. |

Since the first version of Moja Reporting Tool is more geared towards Emissions Reporting and the Flux database is more directly linked to emissions reporting requirements than the stock database, the sections that follow only cover the design of the Flux Database in detail.

### Flux Database

As mentioned previously, the Flux database is primarily concerned with capturing 'facts' regarding fluxes between pools within the FLINT. It is designed as a Star Schema which constitutes a 'Fact' Table that contains flux values which are associated with attributes in a series of 'Dimension' tables. Fact and Dimension tables are linked via a dimension id.



*Figure 2: Flux Database Entity Design*

|  | Flux DB Table | | Description |
| --- | --- | --- | --- |
| 1. | Flux Fact | | The FluxFact table consists of 'records' which have flux fact attributes as well as attributes that link to the dimension tables through unique dimension identifiers. The flux facts include a flux value which is always positive (because it represents the movement of mass from one pool (StartPool) to another (EndPool)). In addition the flux facts have attributes of StartPool and EndPool, which identifies the source pool of the flux and the destination pool of the flux. The StartPool and EndPool can be used to determine how a flux is related to the IPCC carbon stock changes and non-CO2 emissions. |
| 2. | Date dimension | | The date dimension table contains information on the timing of the simulation steps associated with the FluxFacts. The number of date attributes could potentially be as high as the finest time-step scale of the simulation, however, practically this may not be feasible due to database size. As of the current implementation, the Date Dimension is restricted to either monthly or yearly aggregation to manage database size. |
| 3. | Flux Type Info Dimension | | The Flux Type Info Dimension captures the cause of the flux. This information is used to attribute the flux to certain pools and classifications in different reporting outputs, for example, UNFCCC land use classifications and emission sources. |
| 4. | Pool Info Dimension | | The Pool Info dimension captures where a Flux came from (StartPool) and where it went to (EndPool). This information is used to assign fluxes to one of the IPCC carbon stock change pools as well as the non-CO2 emissions |
| 5. | Location Dimension | | The Location Dimension table captures the spatial location information for the FluxFacts. This information is used to aggregate all fluxes that occur on a Simulation Unit with fluxes from other Simulation Units that have the same location Id |
| 6. | County Info Dimension | | The County Info Dimension table captures the details of the county (administrative unit) within which a Flux occurred |
| 7.` | Vegetation History Dimension | | The Vegetation History Dimension table allows FluxFacts to be assigned a land use classification that takes into account the land-use history. This dimension table enables the Approach 3 capability of the FLINT to be implemented in the output database and used in the Reporting Tool. |
| 8. | Vegetation Type Dimension | | The Vegetation Type Dimension table contains additional information about country specific species. It then provides for the mapping of the Vegetation Types to IPCC Cover Types (F=Forest land, C=Cropland, G=Grassland, W=Wetland, S=Settlements, O=Other land), Natural System (Natural System = Yes or No), and a Woodiness status (Woody = Yes or No). |
| 9. | IPCC Cover Type Info Dimension | | Attributes a species to an IPCC category, Forest land, Cropland, Grassland, Wetlands, Settlements, Other land |

## Reporting Tool Database

The Reporting Tool’s Database hosts all the data tables that are needed by the Reporting Tool that are not already a part of the FLINT’s Flux Output Database.

Apache Cassandra was chosen as the database provider of choice for the Reporting Tool because:

1. Being a NoSQL database, it does not make use of blocking JDBC drivers:

This is a major performance boost when it comes to writing processed results to disk.

1. Using its "Gossip Protocol", it allows read and write events to occur on any of its replicas:

This is a major performance boost when it comes to writing processed results to disk.

1. Writes are the cheapest thing you can do on Cassandra:

Since the Reporting Tool write huge amounts of data in a small span of time, this is a huge plus,

Below is a description of the tables contained in the Reporting Tool Database:

|  | Reporting Tool DB Table | | Description |
| --- | --- | --- | --- |
| 1. | Database Event Log | | Stores the details of all lifecycle events that occur during the processing of a Flux Output Database. For example, the start of the processing cycle and the end of the processing cycle are each individually logged in this table for monitoring purposes |
| 2. | Database | | Stores the key details of a Flux Output database that has been submitted for processing, including its name, start year, end year and the current state of processing that it is currently marked as being in. For instance, a newly submitted database will be marked as being in a new state and a processed database will be marked as being in a processed state. |
| 3. | Emission Types | | Store the details of the different types of Greenhouse Gas Emissions. This includes the title, abbreviation and a short description of the Gas. |
| 4. | Flux Aggregation Rules | | Store the instructions of how a flux fact should be treated (added, subtracted, or ignored) for UNFCCC reporting variables (simplified IPCC carbon stock change and emissions pools) depending on where it came from (StartPool) and where it went to (EndPool) |
| 5. | Reporting Table | | Store the details of Draft 2006 GL Reporting Tables based upon the Common Reporting Format Tables agreed at SUBSTA39. This includes the number, name and description of the table |
| 6. | Land Use Classification | | Stores the details of the key land uses classes i.e Forest land, Cropland, Grassland, Wetlands, Settlements and Other lands. This includes the title of the land use class and a reference to the corresponding IPCC Cover Type Id in the Flux Output Database IPCC Cover Type Info Dimension |
| 7. | Land Use Subclassification | | Stores the details of the key land uses subclasses for Land Remaining Land and Land Converted To Land. This includes the title of the land use subclass and a reference to the corresponding Parent Land Use Class (See 6 above) and IPCC Cover Type Id in the Flux Output Database IPCC Cover Type Info Dimension. |
| 8. | Flux Aggregation | | Stores the instructions of the table that a flux fact should be aggregated given its flux type in combination with its land use class |
| 9. | Conversion and Remaining Periods | | Stores the policy rules that need to be considered when applying a Land Use Classification for IPCC Land Use Categories including Years to move to a Remaining or Conversion category. |
| 10. | Emission Type Source | | Stores the list sources to which emission types can be attributed e.g. Controlled Burning, Wildfire or Net carbon stock change in living biomass etc. |
| 11. | County Emissions | | Stores the aggregated emissions - at the county level |
| 12. | National Emissions | | Stores the aggregated emissions - at the national level |

## Reporting Tool’s RESTful Services

The functionality of the Reporting Tool is delivered using a microservice based architecture.

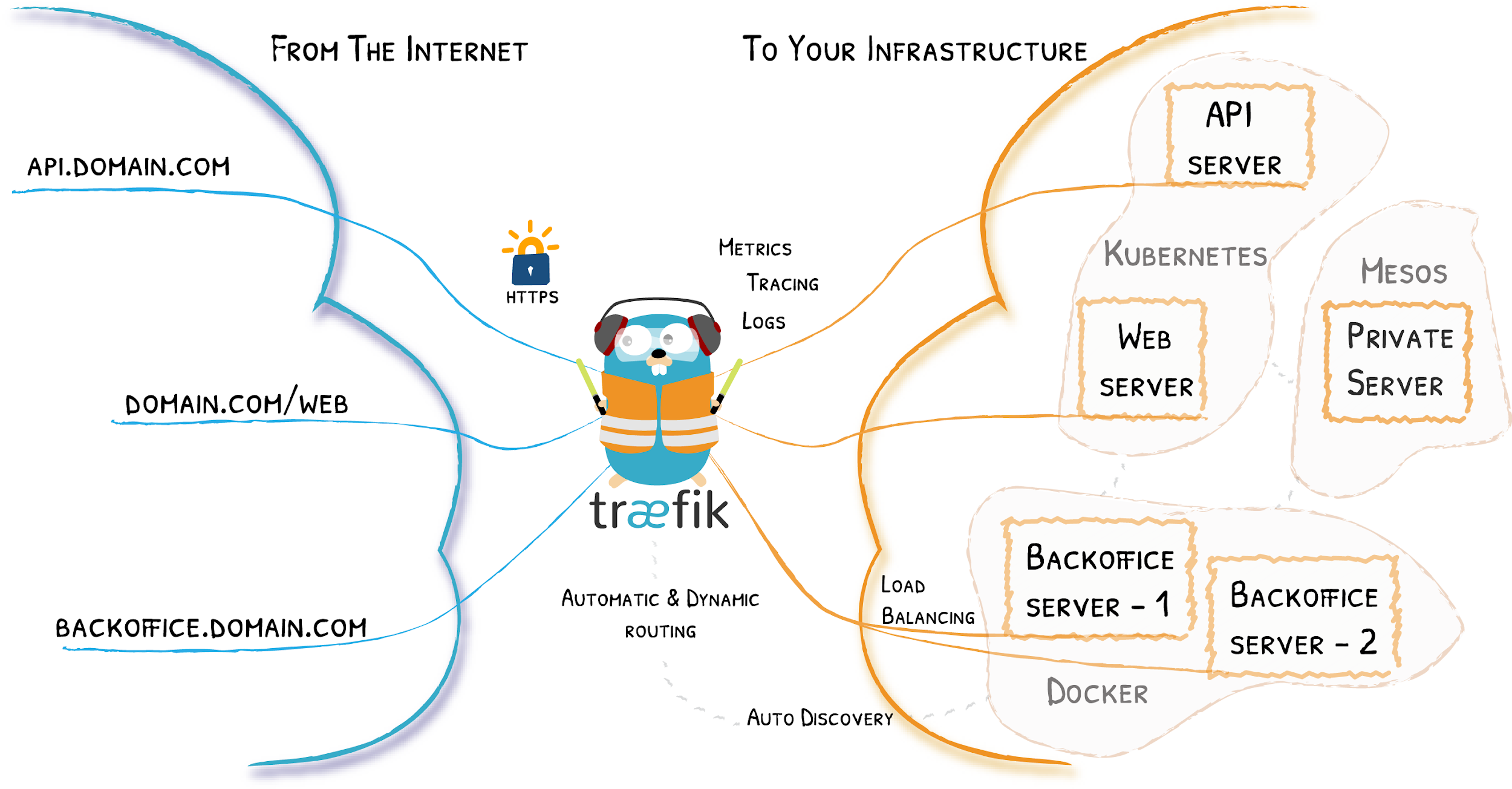
While this was inspired by the many benefits that the architecture provides, the primary motivation was the need to scale different parts of the system independent of each other. This allowed for critical operations of the Tool, such as the one that implements the land use decision tree, to be replicated and assigned more processing resources than the rest of the services, allowing for faster land use classification and thus overall faster processing of newly onboarded Flux Output databases.

Spring Webflux, being a fully non-blocking web stack designed to handle concurrency with a small number of threads and scale with fewer hardware resources was chosen as the framework to implement the Reporting Tool Microservices.

Below is an outline of the microservices that make up the Reporting Tool. Further explanations can be found embedded in the microservices source code.

|  | Microservice | | Description |
| --- | --- | --- | --- |
| 1. | Flux Fact | | RESTful Service that allows GET Requests to be performed against Flux Output Database’s Flux Facts Resource |
| 2. | Date dimension | | RESTful Service that allows GET Requests to be performed against Flux Output Database’s Date dimension Resource |
| 3. | Flux Type Info Dimension | | RESTful Service that allows GET Requests to be performed against Flux Output Database’s Flux Type Info Dimension Resource |
| 4. | Pool Info Dimension | | RESTful Service that allows GET Requests to be performed against Flux Output Database’s Pool Info Dimension Resource |
| 5. | Location Dimension | | RESTful Service that allows GET Requests to be performed against Flux Output Database’s Location Dimension Resource |
| 6. | County Info Dimension | | RESTful Service that allows GET Requests to be performed against Flux Output Database’s County Info Dimension Resource |
| 7.` | Vegetation History Dimension | | RESTful Service that allows GET Requests to be performed against Flux Output Database’s Vegetation History Dimension Resource |
| 8. | Vegetation Type Dimension | | RESTful Service that allows GET Requests to be performed against Flux Output Database’s Vegetation Type Dimension Resource |
| 9. | IPCC Cover Type Info Dimension | | RESTful Service that allows GET Requests to be performed against Flux Output Database’s IPCC Cover Type Info Dimension Resource |
| 10. | Database Event Log | | RESTful Service that allows POST, GET and DELETE Requests to be performed against Reporting Tools Database’s Database Event Logs Resource |
| 11. | Database | | RESTful Service that allows POST, GET, PUT and DELETE Requests to be performed against Reporting Tools Database’s Database Resource. Unlike the other RESTful services, the POST event in this service triggers the Reporting Tool processing exercise which eventually creates the County and National Emissions. |
| 12. | Emission Types | | RESTful Service that allows POST, GET, PUT and DELETE Requests to be performed against Reporting Tools Database’s Emission Types Resource |
| 13. | Flux Aggregation Rules | | RESTful Service that allows POST, GET, PUT and DELETE Requests to be performed against Reporting Tools Database’s Flux Aggregation Rules Resource |
| 14. | Reporting Table | | RESTful Service that allows GET Requests to be performed, that internally read the County and National Emissions Resources and return responses that can be graphed or rendered / downloaded in tabular formats |
| 15. | Land Use Classification | | RESTful Service that allows POST, GET, PUT and DELETE Requests to be performed against Reporting Tools Database’s Land Use Classification Resource |
| 16. | Land Use Subclassification | | RESTful Service that allows POST, GET, PUT and DELETE Requests to be performed against Reporting Tools Database’s Land Use Subclassification Resource |
| 17. | Flux Aggregation | | RESTful Service that allows POST, GET, PUT and DELETE Requests to be performed against Reporting Tools Database’s Flux Aggregation Resource |
| 18. | Conversion and Remaining Periods | | RESTful Service that allows POST, GET, PUT and DELETE Requests to be performed against Reporting Tools Database’s Conversion and Remaining Periods Resource |
| 19. | Emission Type Source | | RESTful Service that allows POST, GET, PUT and DELETE Requests to be performed against Reporting Tools Database’s Emission Type Source Resource |
| 20. | County Emissions | | RESTful Service that allows POST, GET, PUT and DELETE Requests to be performed against Reporting Tools Database’s County Emissions Resource |
| 21. | National Emissions | | RESTful Service that allows POST, GET, PUT and DELETE Requests to be performed against Reporting Tools Database’s National Emissions Resource |

## API Gateway



*Figure 3: Traefik API Gateway Illustration*

All requests to the Reporting Tool Web Services are routed through an API Gateway implemented using Traefik. Traefik is an open-source Edge Router that makes publishing service fun and easy. It receives requests on behalf of the Reporting Tool and finds out which components of the Reporting Tool are responsible for handling them. It then forwards the requests to them and returns the response to the caller.

## Reporting Tool Client

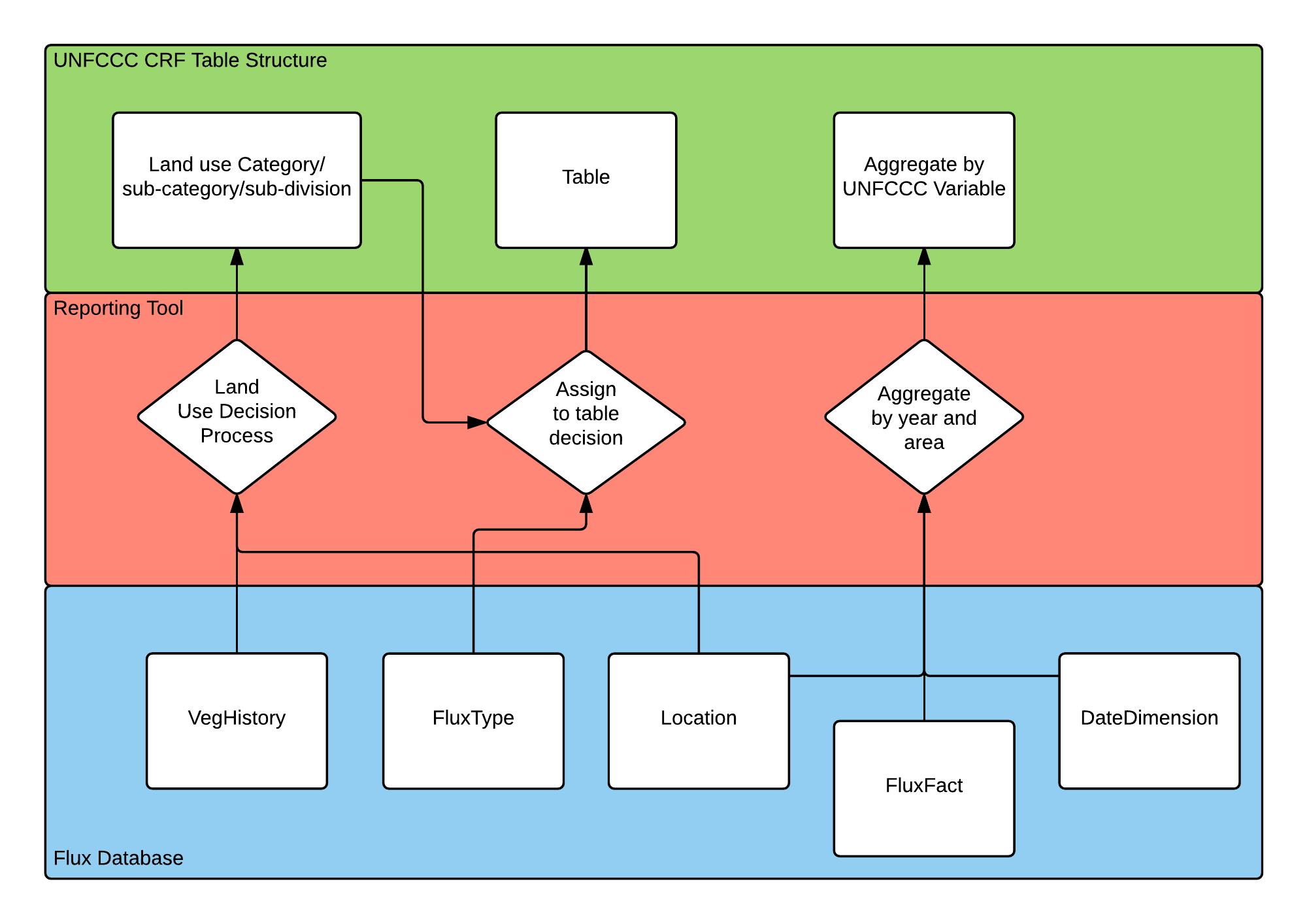
The Reporting Tool’s User interface is implemented using Angular which blends beautifully with the rest of the microservice stack. Primarily, the interface shows emissions in graphical and tabular formats, at counties and national levels, for each of the flux output databases processed by the Reporting Tool. It further allows users to download the tables as neatly formatted xlsx documents for offline reference and reporting.

## 

# Execution Flow

## Overview

The role of the Reporting Tool is to provide Business Intelligence for analysing and transforming FLINT output databases into useful information and outputs. The current implementation of the Reporting Tool processes the Flux database into simplified 2006 GL Reporting Tables . The figure below outlines the three key processes that the Reporting Tool does to achieve this.



*Figure 4: Key Reporting Tool Processes*

The big picture is that the Reporting Tool takes flux facts and assigns / aggregates them to a land use category, a reporting table and a UNFCCC reporting variable.

To do this:

1. The Reporting Tool first implements a process that allocates a flux to an IPCC Land Use Category:

This is identified as the **Land Use Decision Process**.

1. Secondly, it implements a process to allocate the flux to a 2006 GL Reporting Table:

This is identified as the **Flux Type to Reporting Table Process**.

1. Finally, it aggregates the fluxes according to the UNFCCC variable:

This is simply identified as the **Aggregation Process**.

The sections below explore these processes in detail.

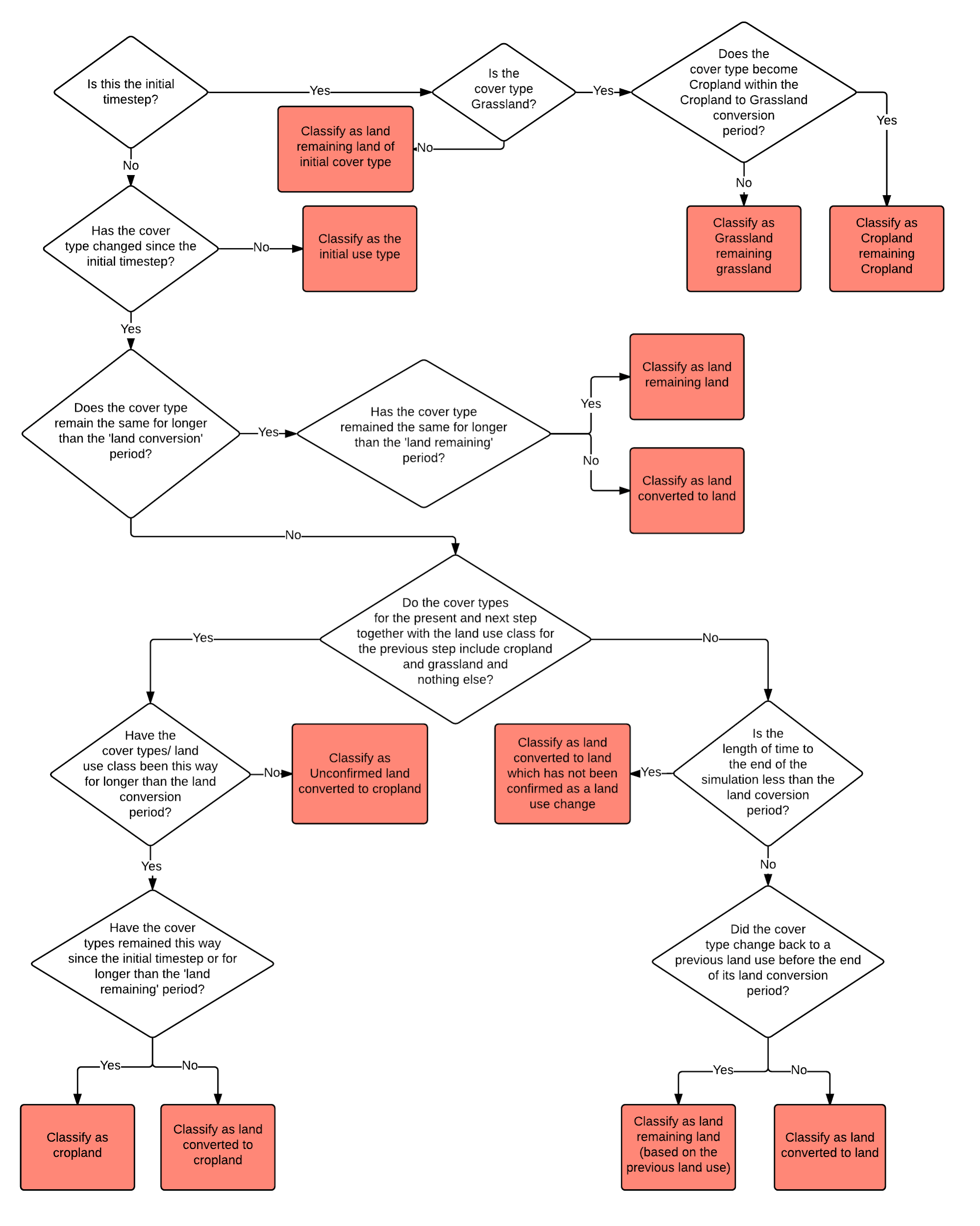
## Land Use Decision Process

The land use decision process is used to assign a UNFCCC land use category for each year for each flux and its associated land area.

The decisions of which IPCC land use category to assign are made based upon the Vegetation History Dimension (VegHistoryDim) and it's associated dimensions in the Flux database . The VegHistoryDim contains information on the vegetation type (VegType) for each year of the simulation, including the initial type. The VegType is associated with an IPCC cover type which is used in the land use decision process, to decide which land use category to assign for each year.

The land use decision process supports all of the IPCC land use categories and their sub-classifications of land use 'remaining' land use and land use 'converted to' land use.

The figure on the next page outlines the Reporting Tool’s land use decision process. The Reporting Tool triggers this process whenever a new Flux Output database is added to the system.



*Figure 5: Reporting Tools Land Use Decision Process*

## 

## Flux Type to Reporting Table Process

The FluxType in combination with the land use classification determines the Reporting Table that a Flux is aggregated. The table below indicates which table the Reporting Tool aggregates Fluxes for each combination of Flux Type, Land Use Class and Emissions Type (CO2 (carbon stock change as well as CO2), CH4 and N2O) .

| id | Flux Type | Land Use Class | Description | CO2 | CH4 | N2O |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Wildfire | Forest land | Fluxes that are the result of a wildfire disturbance event. This can include emissions to the atmosphere as well as transfers between pools and can include carbon and other non-carbon greenhouse gases | 4.A | 4(V) | 4(V) |
| 1 | Wildfire | Cropland | Fluxes that are the result of a wildfire disturbance event. This can include emissions to the atmosphere as well as transfers between pools and can include carbon and other non-carbon greenhouse gases | 4.B | 4(V) or 3.F | 4(V) or 3.F |
| 1 | Wildfire | Grassland | Fluxes that are the result of a wildfire disturbance event. This can include emissions to the atmosphere as well as transfers between pools and can include carbon and other non-carbon greenhouse gases | 4.C | 4(V) or 3.E | 4(V) or 3.E |
| 1 | Wildfire | Wetlands | Fluxes that are the result of a wildfire disturbance event. This can include emissions to the atmosphere as well as transfers between pools and can include carbon and other non-carbon greenhouse gases | 4.D | 4(V) | 4(V) |
| 1 | Wildfire | Settlements | Fluxes that are the result of a wildfire disturbance event. This can include emissions to the atmosphere as well as transfers between pools and can include carbon and other non-carbon greenhouse gases | 4.E | 4(V) | 4(V) |
| 2 | Controlled Burning | Forest land | Fluxes that are the result of a controlled burning disturbance. This can include emissions to the atmosphere as well as transfers between pools | 4.A | 4(V) | 4(V) |
| 2 | Controlled Burning | Cropland | Fluxes that are the result of a controlled burning disturbance. This can include emissions to the atmosphere as well as transfers between pools | 4.B | 4(V) or 3.F | 4(V) or 3.F |
| 2 | Controlled Burning | Grassland | Fluxes that are the result of a controlled burning disturbance. This can include emissions to the atmosphere as well as transfers between pools | 4.C | 4(V) or 3.E | 4(V) or 3.E |
| 2 | Controlled Burning | Wetlands | Fluxes that are the result of a controlled burning disturbance. This can include emissions to the atmosphere as well as transfers between pools | 4.D | 4(V) | 4(V) |
| 2 | Controlled Burning | Settlements | Fluxes that are the result of a controlled burning disturbance. This can include emissions to the atmosphere as well as transfers between pools | 4.E | 4(V) | 4(V) |
| 3 | NPP | Forest land | Net primary production - the mass of photosynthate remaining after taking into account loss due to autotrophic respiration | 4.A | NA | NA |
| 3 | NPP | Cropland | Net primary production - the mass of photosynthate remaining after taking into account loss due to autotrophic respiration | 4.B | NA | NA |
| 3 | NPP | Grassland | Net primary production - the mass of photosynthate remaining after taking into account loss due to autotrophic respiration | 4.C | NA | NA |
| 3 | NPP | Wetlands | Net primary production - the mass of photosynthate remaining after taking into account loss due to autotrophic respiration | 4.D | NA | NA |
| 3 | NPP | Settlements | Net primary production - the mass of photosynthate remaining after taking into account loss due to autotrophic respiration | 4.E | NA | NA |
| 4 | Harvest | Forest land | Fluxes that are the result of a harvest event where all trees are harvested which can be fluxes between onsite pools or can be fluxes to offsite pools | 4.A | NA | NA |
| 4 | Harvest | Grassland | Fluxes that are the result of a harvest event where woody and/or non-woody veg is harvested which can be fluxes between onsite pools or can be fluxes to offsite pools | 4.C | NA | NA |
| 4 | Harvest | Cropland | Fluxes that are the result of a harvest event where woody and/or non-woody veg is harvested which can be fluxes between onsite pools or can be fluxes to offsite pools | 4.B | NA | NA |
| 5 | Decomposition | Forest land | Fluxes to the atmosphere due to the decay of organic matter in dead organic matter and soil | 4.A | NA | 4(III) |
| 5 | Decomposition | Cropland | Fluxes to the atmosphere due to the decay of organic matter in dead organic matter and soil | 4.B | NA | 3.D |
| 5 | Decomposition | Grassland | Fluxes to the atmosphere due to the decay of organic matter in dead organic matter and soil | 4.C | NA | 4(III) |
| 5 | Decomposition | Wetlands | Fluxes to the atmosphere due to the decay of organic matter in dead organic matter and soil | 4.D | NA | 4(III) |
| 5 | Decomposition | Settlements | Fluxes to the atmosphere due to the decay of organic matter in dead organic matter and soil | 4.E | NA | 4(III) |
| 6 | Turnover | Forest land | Fluxes due to normal processes of leaf, bark, and branch shedding and movement through debris and soil carbon pools due to decay | 4.A | NA | NA |
| 6 | Turnover | Cropland | Fluxes due to normal processes of leaf, bark, and branch shedding and movement through debris and soil carbon pools due to decay | 4.B | NA | NA |
| 6 | Turnover | Grassland | Fluxes due to normal processes of leaf, bark, and branch shedding and movement through debris and soil carbon pools due to decay | 4.C | NA | NA |
| 6 | Turnover | Wetlands | Fluxes due to normal processes of leaf, bark, and branch shedding and movement through debris and soil carbon pools due to decay | 4.D | NA | NA |
| 6 | Turnover | Settlements | Fluxes due to normal processes of leaf, bark, and branch shedding and movement through debris and soil carbon pools due to decay | 4.E | NA | NA |
| 7 | Plough | Cropland | Fluxes from a tillage disturbance on crop and grass | 4.B | NA | NA |
| 7 | Plough | Grassland | Fluxes from a tillage disturbance on crop and grass | 4.C | NA | NA |
| 8 | Thin | Forest land | Fluxes that are the result of a thin event where not all trees are harvested or felled which can be fluxes between onsite pools or can be fluxes to offsite pools | 4.A | NA | NA |
| 8 | Thin | Grassland | Fluxes that are the result of a thin event where not all woody veg is harvested or felled which can be fluxes between onsite pools or can be fluxes to offsite pools | 4.B | NA | NA |
| 8 | Thin | Cropland | Fluxes that are the result of a thin event where not all woody veg is harvested or felled which can be fluxes between onsite pools or can be fluxes to offsite pools | 4.C | NA | NA |
| 9 | Plant Trees | Forest land | When tree seedlings are planted there is a small carbon flux from offsite to tree pools | 4.A | NA | NA |
| 10 | Plant Crops | Cropland | When crops are sown or planted there is a carbon flux from offsite to crop pools | 4.B | NA | NA |
| 11 | N Fertiliser | Forest land | A flux that is the result of nitrogen fertiliser application | NA | NA | 4(I) |
| 11 | N Fertiliser | Cropland | A flux that is the result of nitrogen fertiliser application | NA | NA | 3.D |
| 11 | N Fertiliser | Grassland | A flux that is the result of nitrogen fertiliser application | NA | NA | 3.D |
| 11 | N Fertiliser | Wetlands | A flux that is the result of nitrogen fertiliser application | NA | NA | 4(I) |
| 11 | N Fertiliser | Settlements | A flux that is the result of nitrogen fertiliser application | NA | NA | 4(I) |
| 12 | Pest | Forest land | Fluxes due to pest attack, for example transfer of leaves to the fine litter pools. | 4.A | NA | NA |
| 12 | Pest | Cropland | Fluxes due to pest attack, for example transfer of leaves to the fine litter pools. | 4.B | NA | NA |
| 12 | Pest | Grassland | Fluxes due to pest attack, for example transfer of leaves to the fine litter pools. | 4.C | NA | NA |
| 13 | Windthrow | Forest land | fluxes due to tree death due to being blown over by wind | 4.A | NA | NA |
| 14 | Lime Application | Forest land | carbon fluxes associated with lime and dolomite application | 3.G.I | NA | NA |
| 14 | Lime Application | Cropland | carbon fluxes associated with lime and dolomite application | 3.G.I | NA | NA |
| 14 | Lime Application | Grassland | carbon fluxes associated with lime and dolomite application | 3.G.I | NA | NA |
| 14 | Lime Application | Wetlands | carbon fluxes associated with lime and dolomite application | 3.G.I | NA | NA |
| 14 | Lime Application | Settlements | carbon fluxes associated with lime and dolomite application | 3.G.I | NA | NA |
| 15 | Grazing | Cropland | fluxes between plant pools due to animal grazing | 4.B | NA | NA |
| 15 | Grazing | Grassland | fluxes between plant pools due to animal grazing | 4.C | NA | NA |
| 16 | Irrigation | Forest land | H20 fluxes due to irrigation | NA | NA | NA |
| 16 | Irrigation | Cropland | H20 fluxes due to irrigation | NA | NA | NA |
| 16 | Irrigation | Grassland | H20 fluxes due to irrigation | NA | NA | NA |
| 17 | Herbicide | Forest land | fluxes due to plant death from herbicide application | 4.A | NA | NA |
| 17 | Herbicide | Cropland | fluxes due to plant death from herbicide application | 4.B | NA | NA |
| 17 | Herbicide | Grassland | fluxes due to plant death from herbicide application | 4.C | NA | NA |
| 18 | Manure | Forest land | Fluxes from the application of manure from offsite | 4.A | 3.B(a) | 3.B(b) |
| 18 | Manure | Cropland | Fluxes from the application of manure from offsite | 4.B | 3.B(a) | 3.B(b) |
| 18 | Manure | Grassland | Fluxes from the application of manure from offsite | 4.C | 3.B(a) | 3.B(b) |
| 19 | Heavy roll | Forest land | flux of carbon due to chopper rolling | 4.A | NA | NA |
| 20 | Wetland Drainage | Forest land | Flux of CO2 ,CH4 and N2O due to wetland drainage and rewetting | 4(II) | 4(II) | 4(II) |
| 20 | Wetland Drainage | Cropland | Flux of CO2 ,CH4 and N2O due to wetland drainage and rewetting | 4(II) | 4(II) | 3.D |
| 20 | Wetland Drainage | Grassland | Flux of CO2 ,CH4 and N2O due to wetland drainage and rewetting | 4(II) | 4(II) | 3.D |
| 20 | Wetland Drainage | Wetlands | Flux of CO2 ,CH4 and N2O due to wetland drainage and rewetting | 4(II) | 4(II) | 4(II) |
| 21 | Rice | Cropland | Special case where ,CH4 is transmitted through the stem | NA | 3.C | NA |
| 22 | Cattle Grazing | Forest land | carbon fluxes due to cattle grazing | 4.A | NA | NA |
| 22 | Cattle Grazing | Cropland | carbon fluxes due to cattle grazing | 4.B | NA | NA |
| 22 | Cattle Grazing | Grassland | carbon fluxes due to cattle grazing | 4.C | NA | NA |
| 23 | Sheep Grazing | Forest land | carbon fluxes due to sheep grazing | 4.A | NA | NA |
| 23 | Sheep Grazing | Cropland | carbon fluxes due to sheep grazing | 4.B | NA | NA |
| 23 | Sheep Grazing | Grassland | carbon fluxes due to sheep grazing | 4.C | NA | NA |
| 24 | Goat Grazing | Forest land | carbon fluxes due to goat grazing | 4.A | NA | NA |
| 24 | Goat Grazing | Cropland | carbon fluxes due to goat grazing | 4.B | NA | NA |
| 24 | Goat Grazing | Grassland | carbon fluxes due to goat grazing | 4.C | NA | NA |

## Aggregation Process

The FluxFact table in the Flux database captures where a Flux came from (StartPool) and where it went to (EndPool). This makes it possible to assign the flux to one of the IPCC carbon stock change pools as well as to non-CO2 emissions.

The table below indicates how the Reporting Tool treats Flux Fact during aggregation (adds, subtracts, or ignores them) for UNFCCC reporting variables (simplified IPCC carbon stock change and emissions pools). A flux is

always positive (because it represents a movement of mass from one pool to another) and as such it is either added (where it is a flux into the pool in question) or subtracted (where it is a flux from the pool in question).

| **FluxFact Start and End Pool Combinations** | **Net carbon stock change in living biomass**  **(kt C)** | **Net carbon stock change in dead organic matter**  **(kt C)** | **Net carbon stock change in mineral soils (kt C)** | **Net carbon stock change in organic soils (kt C)** | **CH4 (controlled burning and wildfire) (kt CH4)** | **N2O (controlled burning and wildfire) (kg N2O)** |
| --- | --- | --- | --- | --- | --- | --- |
| AgDeadFineRoots:SoilBioF | NA | subtract | add | add | NA | NA |
| AgDeadFineRoots:SoilBioS | NA | subtract | add | add | NA | NA |
| AgDeadFineRoots:SoilDPM | NA | subtract | add | add | NA | NA |
| AgDeadFineRoots:SoilHUM | NA | subtract | add | add | NA | NA |
| AgDeadFineRoots:SoilIOM | NA | subtract | add | add | NA | NA |
| AgDeadFineRoots:SoilRPM | NA | subtract | add | add | NA | NA |
| AgFineLitter:Atmosphere CH4 | NA | NA | NA | NA | add | NA |
| AgFineLitter:Atmosphere N2O | NA | NA | NA | NA | NA | add |
| AgFineLitter:AtmosphereCO2 | NA | subtract | NA | NA | NA | NA |
| AgFineLitter:SoilBioF | NA | subtract | add | add | NA | NA |
| AgFineLitter:SoilBioS | NA | subtract | add | add | NA | NA |
| AgFineLitter:SoilDPM | NA | subtract | add | add | NA | NA |
| AgFineLitter:SoilHUM | NA | subtract | add | add | NA | NA |
| AgFineLitter:SoilIOM | NA | subtract | add | add | NA | NA |
| AgFineLitter:SoilRPM | NA | subtract | add | add | NA | NA |
| AgOffSiteProduct:Atmosphere CH4 | NA | NA | NA | NA | NA | NA |
| AgOffSiteProduct:Atmosphere N2O | NA | NA | NA | NA | NA | NA |
| AgOffSiteProduct:AtmosphereCO2 | NA | NA | NA | NA | NA | NA |
| AgStalkLitter:AgFineLitter | NA | NA | NA | NA | NA | NA |
| AgStalkLitter:Atmosphere CH4 | NA | NA | NA | NA | add | NA |
| AgStalkLitter:Atmosphere N2O | NA | NA | NA | NA | NA | add |
| AgStalkLitter:AtmosphereCO2 | NA | subtract | NA | NA | NA | NA |
| AtmosphereCO2:CropFineRoot | add | NA | NA | NA | NA | NA |
| AtmosphereCO2:CropGrain | add | NA | NA | NA | NA | NA |
| AtmosphereCO2:CropLeaf | add | NA | NA | NA | NA | NA |
| AtmosphereCO2:CropStalk | add | NA | NA | NA | NA | NA |
| AtmosphereCO2:GrassFineRoot | add | NA | NA | NA | NA | NA |
| AtmosphereCO2:GrassLeaf | add | NA | NA | NA | NA | NA |
| AtmosphereCO2:TreeBark | add | NA | NA | NA | NA | NA |
| AtmosphereCO2:TreeBranch | add | NA | NA | NA | NA | NA |
| AtmosphereCO2:TreeCoarseRoot | add | NA | NA | NA | NA | NA |
| AtmosphereCO2:TreeFineRoot | add | NA | NA | NA | NA | NA |
| AtmosphereCO2:TreeFoliage | add | NA | NA | NA | NA | NA |
| AtmosphereCO2:TreeStem | add | NA | NA | NA | NA | NA |
| BioChar:SoilDPM | NA | NA | add | add | NA | NA |
| BioChar:SoilIOM | NA | NA | add | add | NA | NA |
| BioChar:SoilRPM | NA | NA | add | add | NA | NA |
| CropFineRoot:AgDeadFineRoots | subtract | add | NA | NA | NA | NA |
| CropGrain:AgFineLitter | subtract | add | NA | NA | NA | NA |
| CropGrain:AgOffSiteProduct | subtract | NA | NA | NA | NA | NA |
| CropGrain:Atmosphere CH4 | NA | NA | NA | NA | add | NA |
| CropGrain:Atmosphere N2O | NA | NA | NA | NA | NA | add |
| CropGrain:AtmosphereCO2 | subtract | NA | NA | NA | NA | NA |
| CropLeaf:AgFineLitter | subtract | add | NA | NA | NA | NA |
| CropLeaf:AgOffSiteProduct | subtract | NA | NA | NA | NA | NA |
| CropLeaf:Atmosphere CH4 | NA | NA | NA | NA | add | NA |
| CropLeaf:Atmosphere N2O | NA | NA | NA | NA | NA | add |
| CropLeaf:AtmosphereCO2 | subtract | NA | NA | NA | NA | NA |
| CropStalk:AgOffSiteProduct | subtract | NA | NA | NA | NA | NA |
| CropStalk:AgStalkLitter | subtract | add | NA | NA | NA | NA |
| CropStalk:Atmosphere CH4 | NA | NA | NA | NA | add | NA |
| CropStalk:Atmosphere N2O | subtract | NA | NA | NA | NA | add |
| CropStalk:AtmosphereCO2 | subtract | NA | NA | NA | NA | NA |
| Deadwood:Atmosphere CH4 | NA | NA | NA | NA | add | NA |
| Deadwood:Atmosphere N2O | NA | NA | NA | NA | NA | add |
| Deadwood:AtmosphereCO2 | NA | subtract | NA | NA | NA | NA |
| Deadwood:ForestOffSiteProduct | NA | subtract | NA | NA | NA | NA |
| Deadwood:SoilBioF | NA | subtract | add | add | NA | NA |
| Deadwood:SoilBioS | NA | subtract | add | add | NA | NA |
| Deadwood:SoilDPM | NA | subtract | add | add | NA | NA |
| Deadwood:SoilHUM | NA | subtract | add | add | NA | NA |
| Deadwood:SoilIOM | NA | subtract | add | add | NA | NA |
| Deadwood:SoilRPM | NA | subtract | add | add | NA | NA |
| Deadwood:TreeCoarseLitter | NA | NA | NA | NA | NA | NA |
| Deadwood:TreeFineLitter | NA | NA | NA | NA | NA | NA |
| ForestOffSiteProduct:Atmosphere CH4 | NA | NA | NA | NA | NA | NA |
| ForestOffSiteProduct:Atmosphere N2O | NA | NA | NA | NA | NA | NA |
| ForestOffSiteProduct:AtmosphereCO2 | NA | NA | NA | NA | NA | NA |
| GrassFineRoot:AgDeadFineRoots | subtract | add | NA | NA | NA | NA |
| GrassFineRoot:AgOffSiteProduct | subtract | NA | NA | NA | NA | NA |
| GrassLeaf:AgFineLitter | subtract | add | NA | NA | NA | NA |
| GrassLeaf:AgOffSiteProduct | subtract | NA | NA | NA | NA | NA |
| GrassLeaf:Atmosphere CH4 | NA | NA | NA | NA | add | NA |
| GrassLeaf:Atmosphere N2O | NA | NA | NA | NA | NA | add |
| GrassLeaf:AtmosphereCO2 | subtract | NA | NA | NA | NA | NA |
| Manure:SoilBioF | NA | NA | add | add | NA | NA |
| Manure:SoilBioS | NA | NA | add | add | NA | NA |
| Manure:SoilDPM | NA | NA | add | add | NA | NA |
| Manure:SoilHUM | NA | NA | add | add | NA | NA |
| Manure:SoilIOM | NA | NA | add | add | NA | NA |
| Manure:SoilRPM | NA | NA | add | add | NA | NA |
| SoilBioF:Atmosphere N2O | NA | NA | NA | NA | NA | add |
| SoilBioF:AtmosphereCO2 | NA | NA | subtract | subtract | NA | NA |
| SoilBioS:Atmosphere N2O | NA | NA | NA | NA | NA | add |
| SoilBioS:AtmosphereCO2 | NA | NA | subtract | subtract | NA | NA |
| SoilDPM:Atmosphere N2O | NA | NA | NA | NA | NA | add |
| SoilDPM:AtmosphereCO2 | NA | NA | subtract | subtract | NA | NA |
| SoilHUM:Atmosphere N2O | NA | NA | NA | NA | NA | add |
| SoilHUM:AtmosphereCO2 | NA | NA | subtract | subtract | NA | NA |
| SoilHUM:SoilBioF | NA | NA | NA | NA | NA | NA |
| SoilHUM:SoilBioS | NA | NA | NA | NA | NA | NA |
| SoilIOM:Atmosphere N2O | NA | NA | NA | NA | NA | add |
| SoilIOM:AtmosphereCO2 | NA | NA | subtract | subtract | NA | NA |
| SoilRPM:Atmosphere N2O | NA | NA | NA | NA | NA | add |
| SoilRPM:AtmosphereCO2 | NA | NA | subtract | subtract | NA | NA |
| SoilRPM:SoilBioF | NA | NA | NA | NA | NA | NA |
| SoilRPM:SoilBioS | NA | NA | NA | NA | NA | NA |
| SoilRPM:SoilHUM | NA | NA | NA | NA | NA | NA |
| TreeBark:Atmosphere CH4 | NA | NA | NA | NA | add | NA |
| TreeBark:Atmosphere N2O | NA | NA | NA | NA | NA | add |
| TreeBark:AtmosphereCO2 | subtract | NA | NA | NA | NA | NA |
| TreeBark:ForestOffSiteProduct | subtract | NA | NA | NA | NA | NA |
| TreeBark:TreeFineLitter | subtract | add | NA | NA | NA | NA |
| TreeBranch:Atmosphere CH4 | NA | NA | NA | NA | add | NA |
| TreeBranch:Atmosphere N2O | NA | NA | NA | NA | NA | add |
| TreeBranch:AtmosphereCO2 | subtract | NA | NA | NA | NA | NA |
| TreeBranch:Deadwood | subtract | add | NA | NA | NA | NA |
| TreeBranch:ForestOffSiteProduct | subtract | NA | NA | NA | NA | NA |
| TreeBranch:TreeCoarseLitter | subtract | add | NA | NA | NA | NA |
| TreeCoarseLitter:Atmosphere CH4 | NA | NA | NA | NA | add | NA |
| TreeCoarseLitter:Atmosphere N2O | NA | NA | NA | NA | NA | add |
| TreeCoarseLitter:AtmosphereCO2 | NA | subtract | NA | NA | NA | NA |
| TreeCoarseLitter:SoilBioF | NA | subtract | add | add | NA | NA |
| TreeCoarseLitter:SoilBioS | NA | subtract | add | add | NA | NA |
| TreeCoarseLitter:SoilDPM | NA | subtract | add | add | NA | NA |
| TreeCoarseLitter:SoilHUM | NA | subtract | add | add | NA | NA |
| TreeCoarseLitter:SoilIOM | NA | subtract | add | add | NA | NA |
| TreeCoarseLitter:SoilRPM | NA | subtract | add | add | NA | NA |
| TreeCoarseLitter:TreeFineLitter | NA | NA | NA | NA | NA | NA |
| TreeCoarseRoot:Atmosphere CH4 | NA | NA | NA | NA | add | NA |
| TreeCoarseRoot:Atmosphere N2O | NA | NA | NA | NA | NA | add |
| TreeCoarseRoot:AtmosphereCO2 | subtract | NA | NA | NA | NA | NA |
| TreeCoarseRoot:TreeDeadCoarseRoot | subtract | add | NA | NA | NA | NA |
| TreeDeadCoarseRoot:Atmosphere CH4 | NA | NA | NA | NA | add | NA |
| TreeDeadCoarseRoot:Atmosphere N2O | NA | NA | NA | NA | NA | add |
| TreeDeadCoarseRoot:AtmosphereCO2 | NA | subtract | NA | NA | NA | NA |
| TreeDeadCoarseRoot:SoilBioF | NA | subtract | add | add | NA | NA |
| TreeDeadCoarseRoot:SoilBioS | NA | subtract | add | add | NA | NA |
| TreeDeadCoarseRoot:SoilDPM | NA | subtract | add | add | NA | NA |
| TreeDeadCoarseRoot:SoilHUM | NA | subtract | add | add | NA | NA |
| TreeDeadCoarseRoot:SoilIOM | NA | subtract | add | add | NA | NA |
| TreeDeadCoarseRoot:SoilRPM | NA | subtract | add | add | NA | NA |
| TreeDeadFineRoot:Atmosphere CH4 | NA | NA | NA | NA | add | NA |
| TreeDeadFineRoot:Atmosphere N2O | NA | NA | NA | NA | NA | add |
| TreeDeadFineRoot:AtmosphereCO2 | NA | subtract | NA | NA | NA | NA |
| TreeDeadFineRoot:SoilBioF | NA | subtract | add | add | NA | NA |
| TreeDeadFineRoot:SoilBioS | NA | subtract | add | add | NA | NA |
| TreeDeadFineRoot:SoilDPM | NA | subtract | add | add | NA | NA |
| TreeDeadFineRoot:SoilHUM | NA | subtract | add | add | NA | NA |
| TreeDeadFineRoot:SoilIOM | NA | subtract | add | add | NA | NA |
| TreeDeadFineRoot:SoilRPM | NA | subtract | add | add | NA | NA |
| TreeFineLitter:Atmosphere CH4 | NA | NA | NA | NA | add | NA |
| TreeFineLitter:Atmosphere N2O | NA | NA | NA | NA | NA | add |
| TreeFineLitter:AtmosphereCO2 | NA | subtract | NA | NA | NA | NA |
| TreeFineLitter:SoilBioF | NA | subtract | add | add | NA | NA |
| TreeFineLitter:SoilBioS | NA | subtract | add | add | NA | NA |
| TreeFineLitter:SoilDPM | NA | subtract | add | add | NA | NA |
| TreeFineLitter:SoilHUM | NA | subtract | add | add | NA | NA |
| TreeFineLitter:SoilIOM | NA | subtract | add | add | NA | NA |
| TreeFineLitter:SoilRPM | NA | subtract | add | add | NA | NA |
| TreeFineRoot:Atmosphere CH4 | NA | NA | NA | NA | add | NA |
| TreeFineRoot:Atmosphere N2O | NA | NA | NA | NA | NA | add |
| TreeFineRoot:AtmosphereCO2 | subtract | NA | NA | NA | NA | NA |
| TreeFineRoot:TreeDeadFineRoot | subtract | NA | NA | NA | NA | NA |
| TreeFoliage:Atmosphere CH4 | NA | NA | NA | NA | add | NA |
| TreeFoliage:Atmosphere N2O | NA | NA | NA | NA | NA | add |
| TreeFoliage:AtmosphereCO2 | subtract | NA | NA | NA | NA | NA |
| TreeFoliage:ForestOffSiteProduct | subtract | NA | NA | NA | NA | NA |
| TreeFoliage:TreeFineLitter | subtract | add | NA | NA | NA | NA |
| TreePlanting:TreeBark | add | NA | NA | NA | NA | NA |
| TreePlanting:TreeBranch | add | NA | NA | NA | NA | NA |
| TreePlanting:TreeCoarseRoot | add | NA | NA | NA | NA | NA |
| TreePlanting:TreeFineRoot | add | NA | NA | NA | NA | NA |
| TreePlanting:TreeFoliage | add | NA | NA | NA | NA | NA |
| TreePlanting:TreeStem | add | NA | NA | NA | NA | NA |
| TreeStem:Atmosphere CH4 | NA | NA | NA | NA | add | NA |
| TreeStem:Atmosphere N2O | NA | NA | NA | NA | NA | add |
| TreeStem:AtmosphereCO2 | subtract | NA | NA | NA | NA | NA |
| TreeStem:Deadwood | subtract | add | NA | NA | NA | NA |
| TreeStem:ForestOffSiteProduct | subtract | NA | NA | NA | NA | NA |
| TreeStem:TreeCoarseLitter | subtract | add | NA | NA | NA | NA |
| ForestOffSiteProduct:ForestOffSiteProduct | NA | NA | NA | NA | NA | NA |
| AgOffSiteProduct:AgOffSiteProduct | NA | NA | NA | NA | NA | NA |
| SoilDPM:Atmosphere CH4 | NA | NA | NA | NA | add | NA |
| SoilRPM:Atmosphere CH4 | NA | NA | NA | NA | add | NA |
| SoilBioF:Atmosphere CH4 | NA | NA | NA | NA | add | NA |
| SoilBioS:Atmosphere CH4 | NA | NA | NA | NA | add | NA |
| SoilHUM:Atmosphere CH4 | NA | NA | NA | NA | add | NA |
| SoilIOM:Atmosphere CH4 | NA | NA | NA | NA | add | NA |

## 

# Source Code Organization

## Reporting Tool Microservices

| --- project root folder  --- pom.xml  --- Dockerfile  --- build.sh  --- register.sh  --- install.sh  --- uninstall.sh  --- licenseheader.txt  --- README.md  | --- src  | --- main  | --- java  --- Application.java  | --- configurations  --- RoutesConfig.java  | --- handlers  | --- models  | --- repositories  | --- util  | --- resources  --- application.properties  | --- test  | --- java  | --- chart  | --- templates  --- service.yaml  --- deployment.yaml  --- ingress.yaml  --- Chart.yaml  --- values.yaml |
| --- |

Please see the next page for more details on what exactly goes into the folder structure.

|  | **Artifact** | | **Description** |
| --- | --- | --- | --- |
| 1. | Project Root Folder | | |
| 1.1. | pom.xml | | XML file containing information about the service and configuration details used by Maven to build the service. |
| 1.2. | Dockerfile | | Text file containing the instructions needed to create a new container image for the service in question. |
| 1.3. | build.sh | | Shell script containing instructions needed to conveniently build the service source code regardless of the path from which it's called. |
| 1.4. | register.sh | | Shell script containing instructions needed to conveniently build the service’s Docker Image and add it to a Docker Repository regardless of the path from which it's called. |
| 1.5. | install.sh | | Shell script containing instructions needed to conveniently deploy the service's container to a Kubernetes cluster regardless of the path from which it's called. |
| 1.6. | uninstall.sh | | Shell script containing instructions needed to conveniently undeploy the service's container from a Kubernetes cluster regardless of the path from which it's called. |
| 1.7. | licenseheader.txt | | Text file container the license header details that should be copied into each Java source file that is created within the service’s folder by Netbeans IDE |
| 1.8. | README.md | | Markdown file describing the project |
| 2. | Project Root Folder / src / main / java / | | |
| 2.1. | Application.java | | The service’s main class |
| 3. | Project Root Folder / src / main / java / configurations | | |
| 3.1. | RoutesConfig.java | | Java file containing a collection of Router Functions that route incoming requests to the corresponding handler functions when the router function matches |
| 3.2. | \*Config.java | | Any other Java configuration file is by default added to this folder. Typical examples include a HostsConfig file which specifies the paths that a service should use access other service that it depends on; A RabbitConfig file that specifies RabbitMQ access parameters; A CassandraConfig and PostgresConfig file that specifies how to access each database respectively. |
| 4. | Project Root Folder / src / main / java / handlers | | |
| 4.1. | \*Handler.java | | All definitions of functions used to handler HTTP requests, and their corresponding implementations are added to this folder |
| 5. | Project Root Folder / src / main / java / repository | | |
| 5.1 | \*.java | | All entity classes that directly map to a database resource in PostgreSQL or Cassandra are added to this folder |
| 6. | Project Root Folder / src / main / java / repositories | | |
| 6.1 | \*Repository.java | | All java files that contain code required to implement data access layers for various persistence stores are added to this folder |
| 7. | Project Root Folder / src / main / java / util | | |
| 7.1. | \*.java | | All other java files that perform utilitarian functions are added to this folder |
| 8. | Project Root Folder / src / main / resources | | |
| 8.1. | application.properties | | Properties file containing all application level configurations. Typically called in the Application.java file to configure the application on boot up |
| 8.2. | \*.properties | | Property files for passing any other tunable customizations typically to the system configuration files e.g HostsConfig, RabbitMQConfig, PostgresConfig, CassandraConfig etc |
| 9. | Project Root Folder / src / test / java | | |
| 9.1. | \*Test.java | | All system tests files |
| 10. | Project Root Folder / chart / templates | | |
| 10.1. | \*service.yaml | | Helm Chart’s Kubernetes service definition file. A service exposes an application running on a set of Pods as a network service |
| 10.2. | \*deployment.yaml | | Helm Chart’s Kubernetes deployment definition file. A deployment provides declarative updates for Pods and ReplicaSets. The desired state is described in a Deployment, and the Deployment Controller changes the actual state to the desired state at a controlled rate |
| 10.3. | \*ingress.yaml | | Helm Chart’s Kubernetes ingress definition file. An ingress exposes HTTP and HTTPS routes from outside the Kubernetes cluster to services within the Kubernetes cluster. Traffic routing is controlled by rules defined on the Ingress resource |
| 11. | Project Root Folder / chart | | |
| 11.1. | Chart.yaml | | Helm Chart’s name and version information |
| 11.2 | values.yaml | | Helm Chart’s configurable values for the Kubernetes service, deployment and ingress files definitions |

## Reporting Tool Client

| --- project root folder  --- Dockerfile  --- build.sh  --- register.sh  --- install.sh  --- uninstall.sh  --- README.md  | --- src  | --- app  | --- core  | --- data  | --- layout  | --- modules  | --- shared  | --- styles  | --- chart  | --- templates  --- service.yaml  --- deployment.yaml  --- ingress.yaml  --- Chart.yaml  --- values.yaml |
| --- |

Please see the next page for more details on what exactly goes into the folder structure.

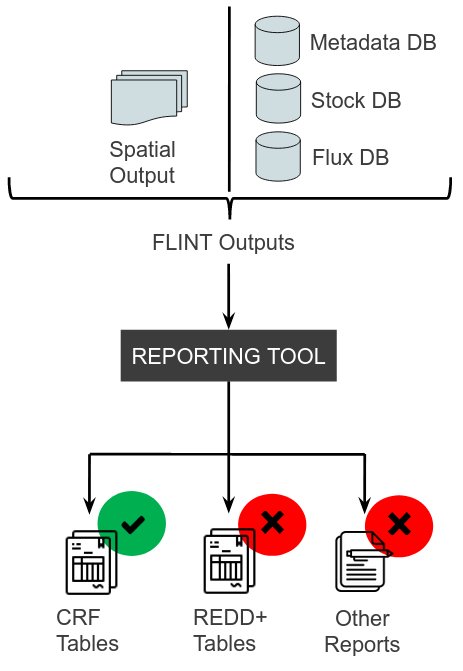
|  | **Artifact** | | **Description** |
| --- | --- | --- | --- |
| 1. | Project Root Folder | | |
| 1.1. | Dockerfile | | Text file containing the instructions needed to create a new container image for the client in question. |
| 1.2. | build.sh | | Shell script containing instructions needed to conveniently build the client source code regardless of the path from which it's called. |
| 1.3. | register.sh | | Shell script containing instructions needed to conveniently build the client’s Docker Image and add it to a Docker Repository regardless of the path from which it's called. |
| 1.4. | install.sh | | Shell script containing instructions needed to conveniently deploy the client's container to a Kubernetes cluster regardless of the path from which it's called. |
| 1.5. | uninstall.sh | | Shell script containing instructions needed to conveniently undeploy the client's container from a Kubernetes cluster regardless of the path from which it's called. |
| 1.6. | README.md | | Markdown file describing the project |
| 2. | Project Root Folder / src / app | | |
| 2.1. | /core | | This app subdirectory is for classes used by app.module. Resources which are always loaded such as route guards, HTTP interceptors, and application level services, such as the ThemeService and logging belong in this directory. |
| 2.2. | /data | | This app subdirectory holds the types (models/entities) and services (repositories) for data consumed by the application |
| 2.3. | /layout | | This app subdirectory holds one or more components which act as a layout or are parts of a layout such as a Header, Nav, Footer, etc. and have a <router-outlet></router-outlet> in the html for other components to embed within |
| 24. | /modules | | This app subdirectory holds a collection of modules which are each independent of each other. This allows Angular to load only the module it requires to display the request thereby saving bandwidth and speeding the entire application |
| 2.5. | / shared | | This app subdirectory holds classes and resources which are used in more than one dynamically loaded module. |
| 3. | Project Root Folder / src / styles | | |
| 3.1. | The ~/src/styles directory is used to store scss style sheets for the application. It can contain themes, Bootstrap, Angular Material, and any other styles. | | |
| 4. | Project Root Folder / chart / templates | | |
| 4.1. | \*service.yaml | | Helm Chart’s Kubernetes service definition file. A service exposes an application running on a set of Pods as a network service |
| 4.2. | \*deployment.yaml | | Helm Chart’s Kubernetes deployment definition file. A deployment provides declarative updates for Pods and ReplicaSets. The desired state is described in a Deployment, and the Deployment Controller changes the actual state to the desired state at a controlled rate |
| 4.3. | \*ingress.yaml | | Helm Chart’s Kubernetes ingress definition file. An ingress exposes HTTP and HTTPS routes from outside the Kubernetes cluster to services within the Kubernetes cluster. Traffic routing is controlled by rules defined on the Ingress resource |
| 5 | Project Root Folder / chart | | |
| 5.1. | Chart.yaml | | Helm Chart’s name and version information |
| 5.2 | values.yaml | | Helm Chart’s configurable values for the Kubernetes service, deployment and ingress files definitions |

# Long Term Vision / Improvements

The Reporting Tool was simply brought together to support the generation of tables, graphs, and other reporting artifacts, from the FLINT output databases, to meet policy and other reporting requirements.

The support for these reporting requirements was not envisioned to happen all at once; but rather in a piece by piece manner, with the first version of the Reporting Tool (the current version) supporting the generation of UNFCCC CRF tables.

Future versions are envisioned to support REDD+ reporting and other reporting requirements.



*Figure 6: Reporting Tools Status and vision for future improvements*

### 

# Logical Next Areas For Improvement

| Issue | Explanation | Action Needed |
| --- | --- | --- |
| Administrative Units | The Reporting Tool code is currently based on the Kenyan Administrative Unit Structure.  The Flux Output Database that served as reference for developing the Reporting Tool was Kenyan Based | A picture of what a generic Flux output database could look like is needed.  Work then needs to be done against this to make the administrative unit and hence reporting system Generic. |
| Data Copying Weaknesses | The Reporting Tool currently has its own PostgreSQL Database instance.  FLINT output is copied from FLINT’s PostgreSQL database instance and imported to the Reporting Tool’s database for processing.  This is a huge problem for a full country spatial run since the database is usually massive | The Reporting Tool should be improved to share the same PostgreSQL database with the FLINT to avoid re-importation issues.  If the above is not possible, the Reporting Tool needs to be redesigned to have its PostgreSQL database in its own optimized server to allow successful importation of dumped data which is memory intensive |
| Data Loading Weaknesses | The FLINT output data is currently loaded into the Reporting Tool’s PostgreSQL database via a script which also kick starts the aggregation exercise | An interface for onboarding data needs to be provided to make it easier for non technical users to operate the system |
| Missing Configuration Screens | The Reporting Tool Report Parameters were fixed based on Kenyan policies.  E.g. The Periods that it takes to have a land conversion were fixed to Kenya’s land use policies | The system’s configurations need to be exposed to allow future users to tweak them to their country requirements |