RE4DY

MANUFACTURING DATA NETWORKS

RE4DY TOOLKIT

Name of the Tool	DAVE - Data Analytics and Visualization Environment
Tool Owner	Industry Commons Foundation
Version	1.0
Date	Nov 2025
Version	V1.0

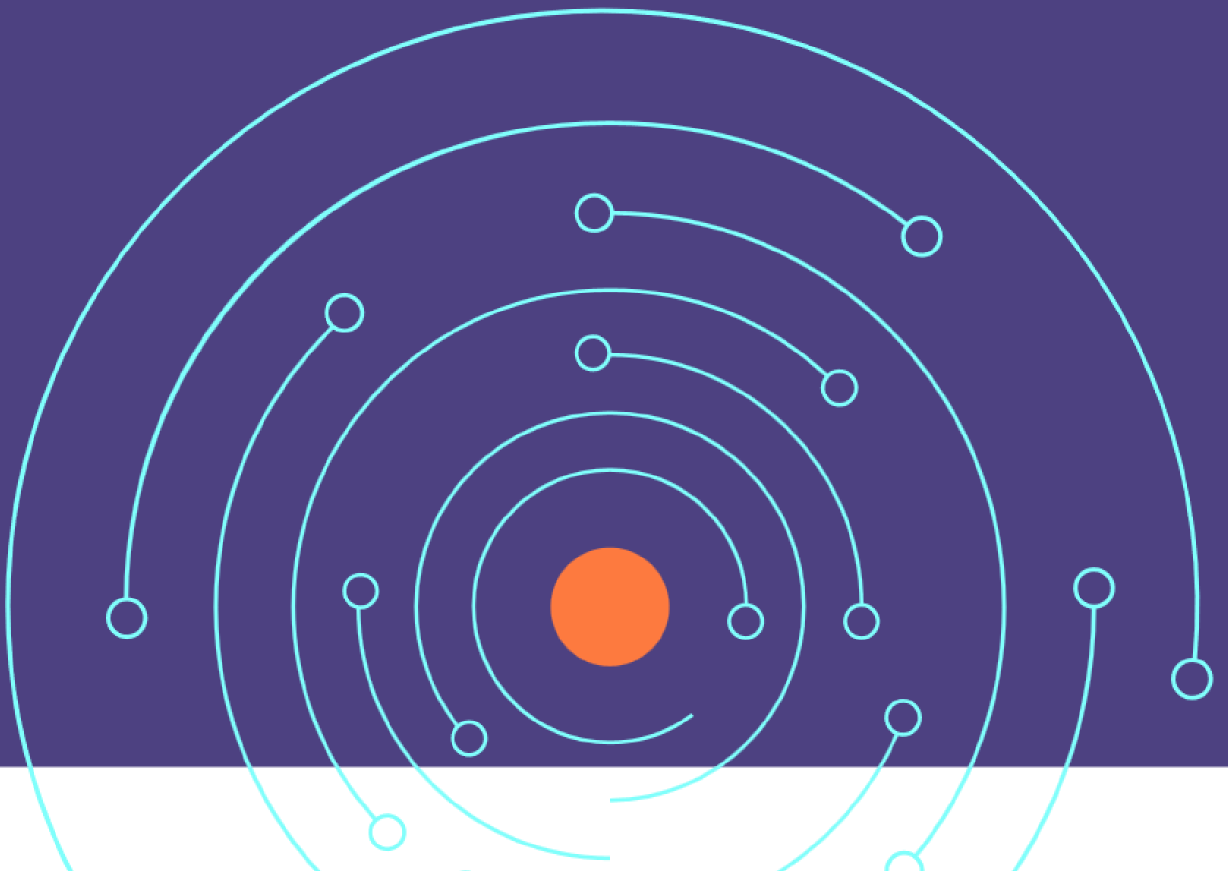


Table of contents

1. Component Description

The Data Analytics and Visualization Environment allow the creation of data analytics workflows in a dynamic way. This solution enables the creation of AI workflows in a simple and intuitive, code-free manner, building workflows using a visual programming environment to place the components through drag-and-drop interface.

The Data Analytics and Visualization Environment solution comprises four major components. The first one is the Apache Airflow platform, responsible for the orchestration of the workflows, meaning the scheduling, execution, monitoring, and storage of the workflows. The second component is the set of supporting technologies, which encompasses all the necessary supporting technologies for workflow creation, as well as AI, Data Analytics, and machine learning algorithms. The third component is the set of operators (connectors, transformers and analytics) used to create workflows, and finally, the last component is the User Interface (UI), which guides users through the dynamic definition of workflows, by enabling the creation of workflows that can connect to data sources (data source configuration, connection and storage), perform data preparation and pre-processing (data filtering, aggregation, harmonisation and semantic enrichment) and apply AI methods for AI model training, updating and serving and Data Analytics (selection and configuration of AI methods).

In the Re4dy project this tool is used to create and automate data integration tasks and create and automate analytic optimization tasks as well as provide data visualization capabilities.

2. Input

The Data Analytics Environment requires the selection of operators from the user in order to form the desired workflows through the user guided interface.

The tool is equipped with multiple operators to support the input of datasets in multiple formats like Excel or CSV files, databases, or message brokers.

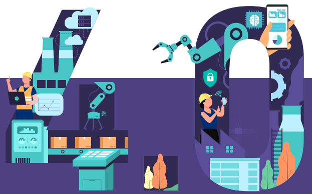
3. Output

Workflows that can be saved to be re-utilized or edited in the future.

Trained AI models - Models that were trained in the workflows.

Modified Datasets - Datasets resulting from the transformations applied in the workflows.

Dashboards – Data dashboards can be created to visualize data as well as results from the workflows.



4. Information Flow

Workflow creation example

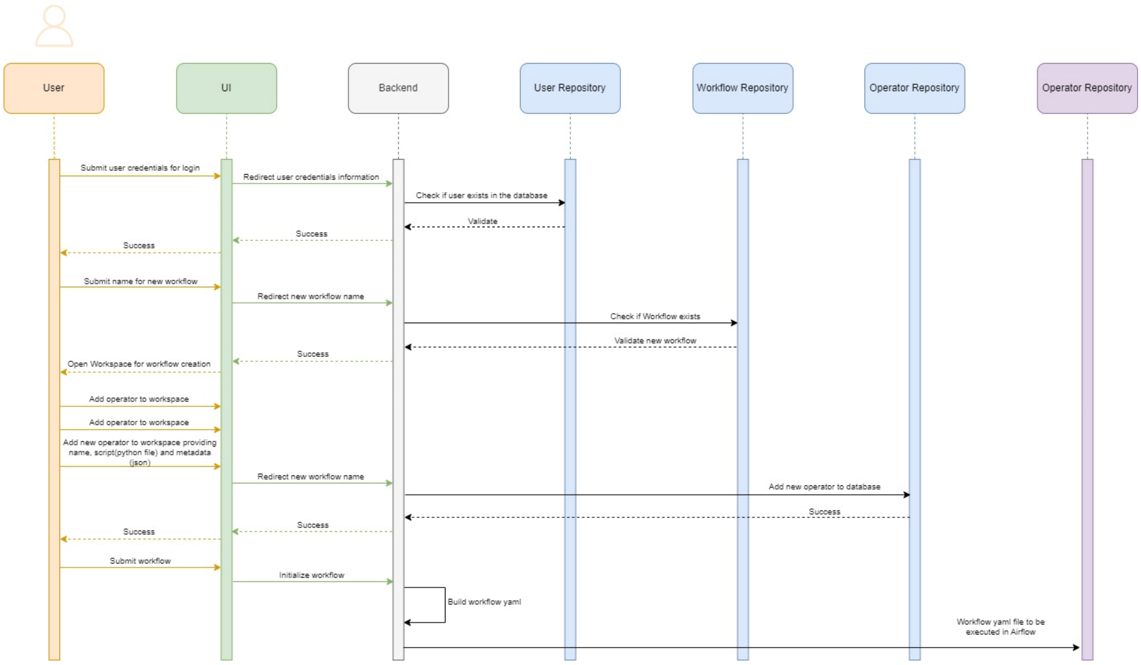

User
UI
Backend
User Repository
Workflow Repository
Operator Repository
Operator Repository
Submit user credentials for login
Redirect user credentials information
Check if user exists in the database
Validate
Success
Success
Submit name for new workflow
Redirect new workflow name
Check if Workflow exists
Validate new workflow
Success
Open Workspace for workflow creation
Add operator to workspace
Add operator to workspace
Add new operator to workspace providing name, script(python file) and metadata (json)
Redirect new workflow name
Add new operator to database
Success
Success
Success
Submit workflow
Initialize workflow
Build workflow yaml
Workflow yaml file to be executed in Airflow

Figure 1 DAVE information flow

5. Internal Architecture

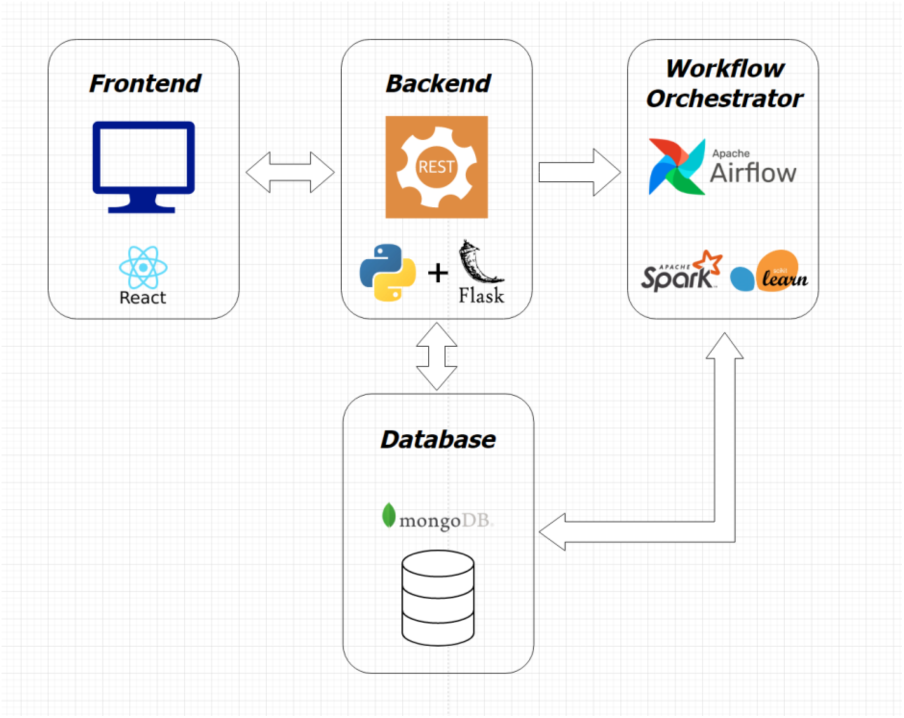


Figure 2 DAVE Architecture

6. API

The internal API was designed to be used by our frontend and integration with other tools. It's not supposed to be used by outside connections.

Table 1. API Endpoint: Create Workflow

Create workflow	
Description	Creates an empty workflow JSON file, where the operators will then be added.
HTTP Method	POST
Endpoint URL	http://host:port/analytics/create_workflow
Parameters	Check **¡Error! No se encuentra el origen de la referencia.**
Outputs	HTTP Response Code: 201 Created

Table 2. API Endpoint: Update Workflow

Update workflow	
Description	Edits the parameters of the chosen workflow, such as the starting date of the execution, the frequency of the execution and other parameters.
HTTP Method	POST
Endpoint URL	http://host:port/analytics/update_workflow
Parameters	Check **¡Error! No se encuentra el origen de la referencia.**
Outputs	HTTP Response Code: 201 Updated

Table 3. API Endpoint: Initialise Workflow Execution

Initialise workflow execution	
Description	Converts the workflow JSON file, with the added operators, to a YAML file, initializes the workflow in Airflow, using the DAG factory library, and saves the workflow to an intermediate storage database.
HTTP Method	POST
Endpoint URL	http://host:port/analytics/init_workflow
Parameters	Check **¡Error! No se encuentra el origen de la referencia.**
Outputs	HTTP Response Code: 201 Initiated -> Airflow UI

Table 4. API Endpoint: Get Workflow Metadata

Get workflow metadata	
Description	Gets the JSON metadata for a workflow. This metadata contains all the accepted inputs for the fields required for the creation of the DAG workflow.
HTTP Method	GET
Endpoint URL	http://host:port/analytics/get_workflow_metadata
Parameters	N.A.
Outputs	HTTP Response Code: 200 OK -> Check **¡Error! No se encuentra el origen de la referencia.**

Table 5. API Endpoint: Delete Workflow

Delete workflow	
Description	Deletes a workflow from the intermediate storage database.
HTTP Method	POST
Endpoint URL	http://host:port/analytics/delete_workflow
Parameters	Workflow Name
Outputs	HTTP Response Code: 201 Deleted

Table 6. API Endpoint: Add Operator

Add operator	
Description	Adds an operator to the current workflow according to the input parameters. Gets the operator type, all the necessary parameters for that operator type and the workflow where the operator will be added.
HTTP Method	POST
Endpoint URL	http://host:port/analytics/add_operator
Parameters	Workflow name Check **¡Error! No se encuentra el origen de la referencia.**
Outputs	HTTP Response Code: 201 Added

Table 7. API Endpoint: Get Operator Metadata

Get operators metadata	
Description	Gets the metadata for all the available operators. This metadata contains all the parameters that each operator accepts, as well as all the accepted inputs.
HTTP Method	GET
Endpoint URL	http://host:port/analytics/get_operators_metadata
Parameters	N.A.
Outputs	HTTP Response Code: 200 OK -> Check **¡Error! No se encuentra el origen de la referencia.**

Table 8. API Endpoint: Delete Operator

Delete operator	
Description	Deletes a selected Operator from a given workflow.
HTTP Method	POST
Endpoint URL	http://host:port/analytics/delete_operator
Parameters	Operator Name, Workflow Name
Outputs	HTTP Response Code: 201 Deleted

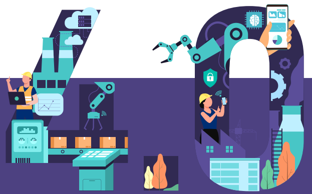
Furthermore, Airflow also provides a RESTFul API that can be used, for instance, to trigger workflows on demand. The Airflow REST API can be used in a number of situations by users, such as in the case of triggering a new workflow execution or listing the set of available workflows for a particular user. The Airflow REST API documentation is available through the following link: https://airflow.apache.org/docs/apache-airflow/stable/stable-rest-api-ref.html.

7. Implementation Technology

The Data Analytics and Visualization Environment is distributed in Linux containers. The backend consists of the Apache airflow backend and a Python Flask API for the different operators. The frontend was developed using the React framework.

The available operators use multiple libraries and frameworks and database connections:

- Keras
- Pandas
- scikit-learn



- Apache Superset
- Apache Spark
- Apache Kafka
- Postgresql
- MongoDB
- MSSQL
- MonetDB

8. Comments

None.