

Final Report for the IEA/ETSAP Project

A TIMES/MIRO App

CONTRACT NO. RC-2021-IFE-001

Project team:

**Frederik Fiand, Robin Schuchmann, Michael Bussieck
GAMS, Germany**

**Evangelos Panos
PSI, Switzerland**

**Markus Blesl
IER, Germany**

**Gary Goldstein
DWI, USA**

February 2022

Foreword

This is a final report that describes the developments of the TIMES/MIRO App that have been accomplished in the scope of the IEA/ETSAP project A TIMES/MIRO App (CONTRACT NO. RC-2021-IFE-001). It is **not** a comprehensive documentation of the TIMES/MIRO App. For the TIMES/MIRO App documentation we refer to the [app_README](#).

Introduction	4
Usage of the TIMES/MIRO App	6
New Features	7
Data Import	7
Settings Tab	8
Improved Data Handling and Charting Facilities	9
RES Viewer	11
Asynchronous Execution of Jobs	12
Scenario Comparison Modes	13
TIMES Cloud Service Backend	15
Implementation Details	16
Documentation	16

1. Introduction

The energy systems analysis community is moving towards open modeling regarding the toolchestand data. An open-source interface to organize runs and analyze inputs and results will contribute to the open-source character of the overall TIMES modeling framework.

This project aims at developing an open-source interface for data input, model solving and results' analysis for the TIMES modeling framework that supports the deployment and use of TIMES models. The TIMES/MIRO app is based on the deployment framework GAMS MIRO¹, which allows building intuitive and user-friendly interfaces that enable efficient interaction with GAMS models and provides extensive visualization capabilities such as charts, time-series, maps, etc.

The TIMES/MIRO app can serve as an open-source GUI to TIMES models as it is published under an open-source license on [GitHub](#). The TIMES/MIRO app can be installed locally or on a MIRO Server and be accessible through the web. When the TIMES/MIRO app is locally installed, a free GAMS demo license is sufficient to use the App and solve large-scale models through the TIMES Cloud Service².

Hence, the TIMES/MIRO App is an interface that provides ETSAP community the possibility to outreach new users of TIMES in developing countries, where budgeting constraints can be a limitation, or to allow potential interested users in developed countries to try and explore the framework before they decide to move to the more sophisticated VEDA interface. The TIMES/MIRO App becomes an important outreach and dissemination tool for TIMES.

Another important feature of TIMES/MIRO App is that the data input and data output interface is based on GAMS and Python and its code is fully accessible, as it is open-source. This means that the TIMES modellers can customise the TIMES/MIRO App as needed. For example, the TIMES modellers using the App could avoid exposing the full input parameters of the model and present only a subset of them that the users of the models are allowed to change. Similarly, on the output side, the users of the App could create sophisticated views of the basic output parameters of TIMES to provide a more user-friendly and at a higher level presentation of results. For example, instead of letting the user of their model selecting "VAR_COMNET" and then "CO2" to see the emissions, TIMES modellers could through the App code, create a new output parameter "Annual_CO2_emissions" which is then available to the users of their model. Therefore, the open source character of the application allows creating *tailor-made* applications for TIMES models that can be a powerful dissemination tool for TIMES-based models to stakeholders or clients and opens new possibilities for the ETSAP community in aiding informed decisions for policymaking.

¹ GAMS MIRO (<https://www.gams.com/miro/>) is a deployment system to transform GAMS models into so-called MIRO applications that run in common web browsers. MIRO apps allow modifying input data, execute GAMS jobs and browse results.

² The TIMES Cloud Service is an ongoing ETSAP R&D undertaking. The service comprises the Open Source TIMES source code, computing resources, and a GAMS license including GAMS/Cplex and GAMS/Conopt. The service is based on GAMS/Engine (<https://www.gams.com/engine/>) that communicates via a REST API. At the current time the **TIMES Cloud Service is only accessible to ETSAP Partners** and their designated representatives.

This report focuses on enhancements and extensions that have been implemented for the underlying prototype³ in the scope of the IEA/ETSAP Project “A TIMES/MIRO App” (CONTRACT NO. RC-2021-IFE-001).

All enhancements and extensions have been tested with a collection of TIMES models that are publicly available (like the [TIMES Demo](#) and the [TIMES-DK COMETS](#) models) or have been provided by the project partners.

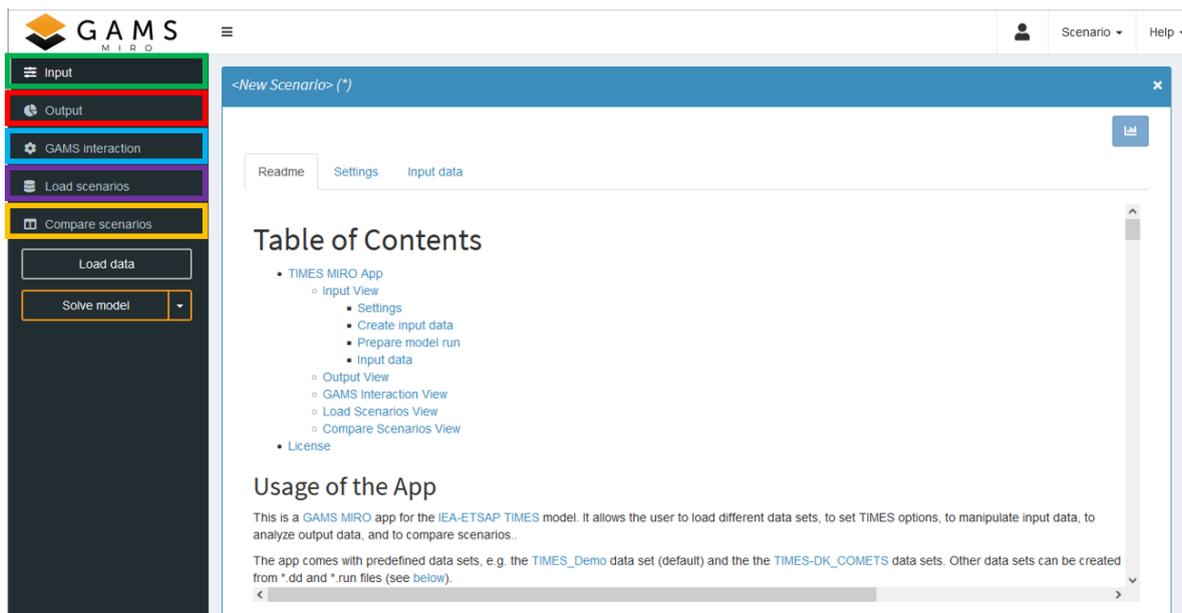
³ GAMS has been working with Evangelos Panos (PSI) and Gary Goldstein (DWI) on a TIMES/MIRO App before this project started. This first prototype of this app has been presented at the July 2020 ETSAP workshop.

2. Usage of the TIMES/MIRO App

The TIMES/MIRO App allows the user to load various data sets, to set TIMES options, to manipulate input data, to analyze output data, and to compare scenarios. The app comes with predefined data sets, e.g. the [TIMES Demo](#) data set (default) and the [TIMES-DK COMETS](#) data sets. Other data sets can be created from *.dd and *.run files produced from VEDA.

The app follows the well-established [MIRO App Structure](#) with a navigation bar on the left side to switch between the following views:

- **Input View**
- **Output View**
- **GAMS Interaction View**
- **Load Scenarios View**
- **Compare Scenarios View**



The navigation bar also contains a Solve button that initiates a run based on settings and data specified in the Input View.

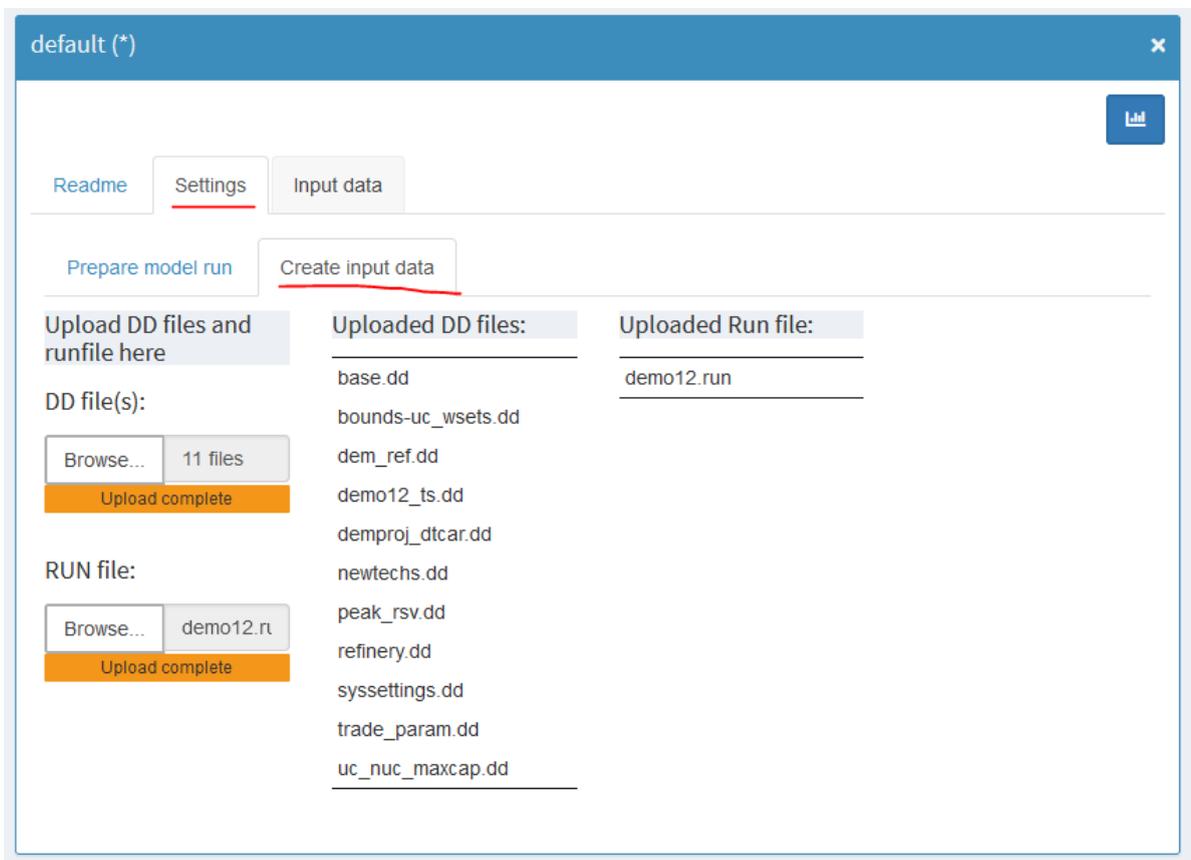
Further details on how to use the TIMES/MIRO App are explained in the [app README](#).

3. New Features

The following features have been implemented in the scope of this project and aim to improve the TIMES/MIRO App user experience.

3.1. Data Import

Data can be easily edited inside the TIMES/MIRO App via a powerful input data cube with pivot table features. While the built-in data editing functionality is well suited to vary small amounts of data quickly, larger data sets can be edited outside the App and imported as a complete set of *.dd files together with a *.run file as produced by VEDA. To make this more convenient, a “Create input data” tab has been introduced in the scope of this project where data files can be provided via drag and drop.



Some additional steps are required to finally import the input data cube and the settings into the TIMES/MIRO App. A detailed description can be found in the [app README](#).

3.2. Settings Tab

The settings tab has been completely refactored in the scope of this project. It manages various widgets allowing users for example to enable/disable TIMES extensions, to select the solver to be used and to specify corresponding solver options, or to set a time limit for the solution.

Readme
Settings
Input data

Prepare model run
Create input data

DD Files order / Read \$offEps ! +

Order (0=ignore)	DD File	\$offEps
1	base	<input type="checkbox"/>
2	newtechs	<input type="checkbox"/>
3	trade_param	<input type="checkbox"/>
4	dem_ref	<input type="checkbox"/>
5	syssettings	<input type="checkbox"/>
6	peak_rsv	<input type="checkbox"/>
7	refinery	<input type="checkbox"/>
8	demproj_dtcar	<input type="checkbox"/>
9	uc_nuc_maxcap	<input type="checkbox"/>
10	bounds-uc_wsets	<input type="checkbox"/>

Extensions +

Extension	Value
REDUCE	YES
DSCAUTO	YES
VDA	YES
DEBUG	NO
DUMPSOL	NO
SOLVE_NOW	YES
XTQA	YES
VAR_UC	YES
SOLVEDA	YES
DATAGDX	YES
VEDAVDD	YES

Solver to use

cplex

Time limit for solve [seconds]

1000

Objective function formulation

MOD

Basis indicator (bRatio) !

0
1

Years for model run +

Year
2005
2010
2015
2020
2030
2050

Time slices available

Time	Value
BOTIME	1960
EOTIME	2200

Time Slice
ANNUAL
S
W
SD
SN
WD
WN

Solver options +

Solver	Option	Value
cplex	scaind	0
cplex	rerun	YES
cplex	iis	YES
cplex	lpmethod	4
cplex	baralg	1
cplex	barcrossalg	1
cplex	barorder	2
cplex	threads	8

3.3. Improved Data Handling and Charting Facilities

The TIMES/MIRO App shows input and output data in powerful Pivot tables that allow slicing and dicing the data as needed. Within the scope of this project, the input data pivot tables have been made editable which allows convenient manipulation of input data by simply double clicking on the corresponding cell.

Readme Settings **Input data**

Region Year Process Commodity Time Slice Limit Types Currencies User Constraint Show All

Filter

Commodity
Symbol
Type

Commodity: CO2 x

Symbol: COM_BDNNET x

Type: Par x

Aggregate: Sum

Load view

Table

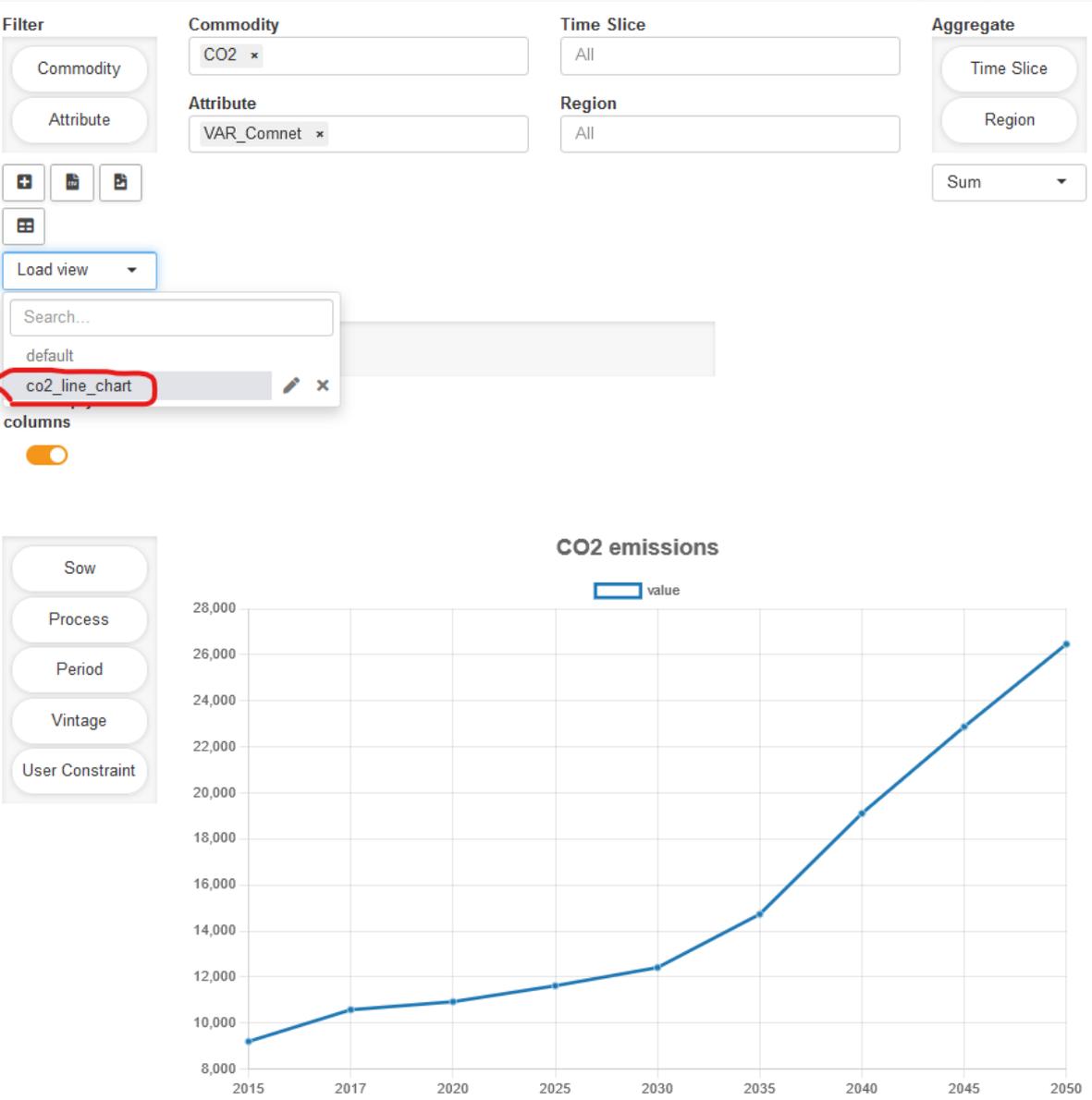
Hide empty columns

Add row Remove rows Search:

DD File	Region	Year	Time Slice	Limit Types	value
pol-co2-20	Starter	2015	ANNUAL	UP	9,198.44
		2017	ANNUAL	UP	10576.256322
		2020	ANNUAL	UP	10,378.47
		2025	ANNUAL	UP	10,746.81
		2030	ANNUAL	UP	11,168.05
		2035	ANNUAL	UP	12,886.75

DD File
User Constraint
Region
Year
Process
Time Slice

Furthermore, new chart types have been added and views that are frequently needed can be saved under a custom name like in the following example where a line chart visualizes CO2 emissions over time.



3.4. RES Viewer

In the upper right corner of the Input Data Tab, users find a button that allows them to switch between the input data cube shown in a pivot table and the newly implemented RES network visualization.

The screenshot shows the RES Viewer interface. At the top right, there is a red-bordered button with a network icon. Below it, the 'Input data' tab is active. The interface includes tabs for 'Process', 'Commodity', and 'UC'. A 'Select commodity' dropdown menu is set to 'AGROILDSL'. The main area displays a network diagram where 'XAGROILDSL00' (yellow) points to 'AGROIL DSL' (white), which then points to a stack of nodes: 'ATHOILDSL-BG', 'ATHOILDSL-BG-X0', 'ATHOILDSL-LT', 'ATHOILDSL-LT-X0', 'AWPOILDSL-IM', 'AWPOILDSL-ST', and 'AWPOILDSL-X0' (all blue). Below the diagram are 'PRE' and 'DMD' buttons. To the right, a table shows details for 'AGROILDSL':

Scenario	base, nt-agr
Type	NRG
SubType	
Activity	PJ
Unit	
TSLVL	
LimType	
PeakTS	
Region	Starter

At the bottom, a pivot table is displayed with the following data:

Table		Sum	Year	
Symbol	Region	value	2015	Totals
-			2.00	2.00
IMPEXP			1.00	1.00
Starter		74.11	3.00	77.11
Totals		74.11	6.00	80.11

The RES viewer provides in itself views that are

- process centric
- commodity centric
- user constraint centric

All displayed items are clickable which allows convenient switching between different views. The corresponding table at the right and the pivot table at the bottom are updated automatically and show related data.

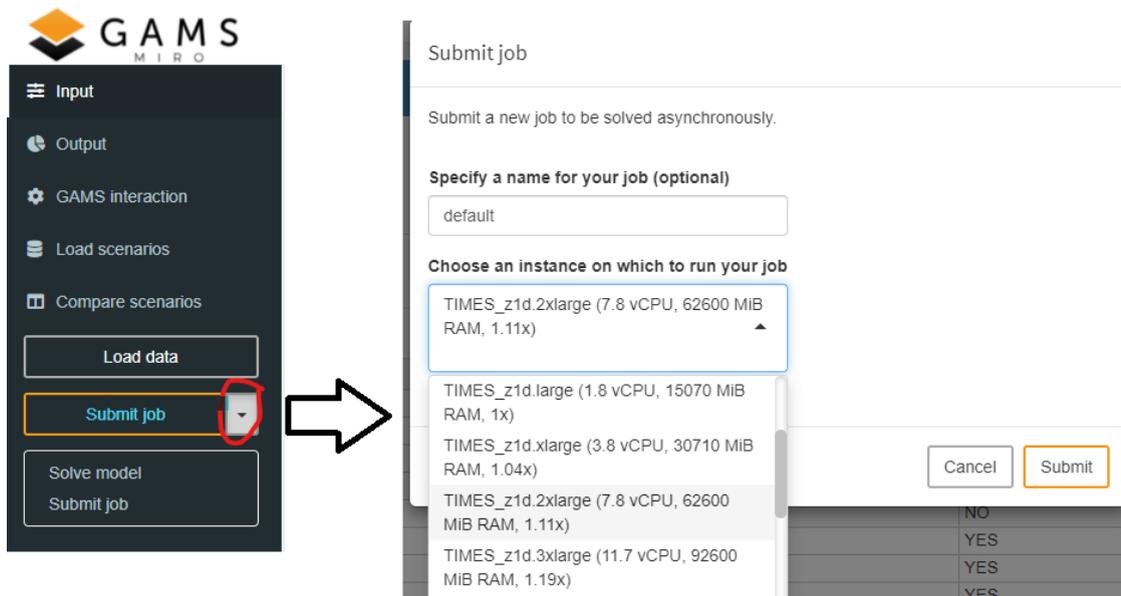
3.5. Asynchronous Execution of Jobs

Jobs can either be executed

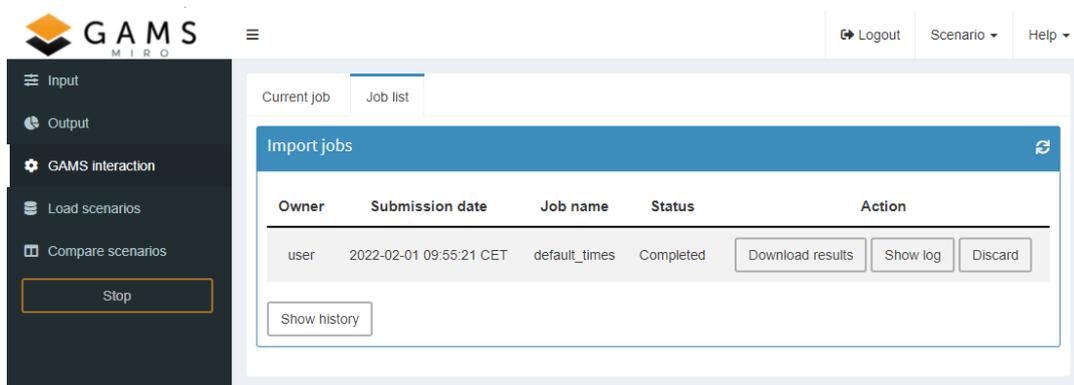
- synchronously (wait for the job to finish, watch the log while the job is running, output data updated automatically) or
- asynchronously (submit the job to the TIMES cloud service and collect the results later)

The asynchronous submission has been added to the TIMES/MIRO App in the scope of this project and is particularly useful, if multiple (computationally expensive) jobs should be executed simultaneously. It requires a GAMS Engine backend (like for example the TIMES Cloud Service) and can be used from a local GAMS MIRO installation or through GAMS MIRO Server (like e.g. <https://times.gams.com/>).

The little arrow in the navigation allows switching between synchronous (“Solve model”) and asynchronous execution (“Submit Job”). When clicking on “Submit Job” a popup occurs that shows the different instance types that are available through the corresponding Engine backend.



Once the user clicks on “Submit”, the job is sent to GAMS Engine for execution. The user can see a list of submitted jobs in the “GAMS interaction” view (“Job list” tab) and download the results once the job has been completed.



3.6. Scenario Comparison Modes

Within the scope of this project, the scenario comparison modes have been extended. Via the “Load Scenarios” view, sophisticated search and filtering mechanisms allow to find the desired scenarios in the database and load multiple scenarios for comparison.

The screenshot shows the 'Load scenarios' interface. On the left is a dark sidebar with navigation options: Input, Output, GAMS interaction, Load scenarios (highlighted), and Compare scenarios. The main area has a blue header 'Load scenarios' and a search section with a condition: Name contains DK_. Below are buttons for 'Add block' and 'Fetch results', and a 'Show 10 entries' dropdown. A table lists three scenarios:

Owner	Name	Time created	Tags	First year available (BO TIME)	Basis indicator	Last year available (EO TIME)	Objective function formulation	Time limit for solve	Location file (for initializ
user	DK_Alternativet	2/1/2022, 9:46:00 AM		1,970.00	1.00	2,200.00	AUTO	3,600.00	/DCCC
user	DK_CB_20	11/12/2021, 12:06:55 PM		1,970.00	1.00	2,200.00	AUTO	1,000.00	/CB_20
user	DK_Regeringen	2/1/2022, 9:47:46 AM		1,970.00	1.00	2,200.00	AUTO	3,600.00	/Social

Once a set of scenarios has been chosen, a popup is offering several ways of comparison

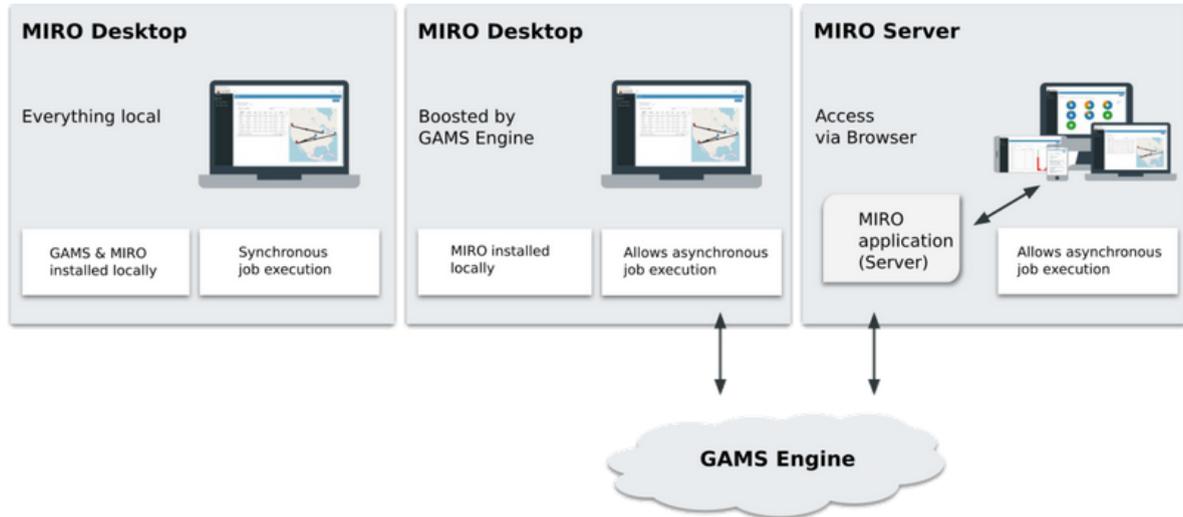
The screenshot shows a 'Select action' popup dialog. It asks 'What would you like to do with the selected scenarios?' and 'Select columns for naming scenarios'. A text input field contains 'Name'. Below are buttons for 'Download data (GDX)' (with a dropdown arrow) and 'Compare (Pivot)'. At the bottom right are 'Cancel' and 'Delete scenarios' buttons.

The newly introduced Pivot view allows slicing and dicing of multiple scenarios' results and therefore convenient comparison between multiple scenarios.



3.7. TIMES Cloud Service Backend

As mentioned before, the GAMS MIRO Apps can be used in different ways from entirely local (“MIRO Desktop”) to entirely online (“MIRO Server”) with a hybrid mode (“MIRO Desktop boosted by GAMS Engine”) in between.



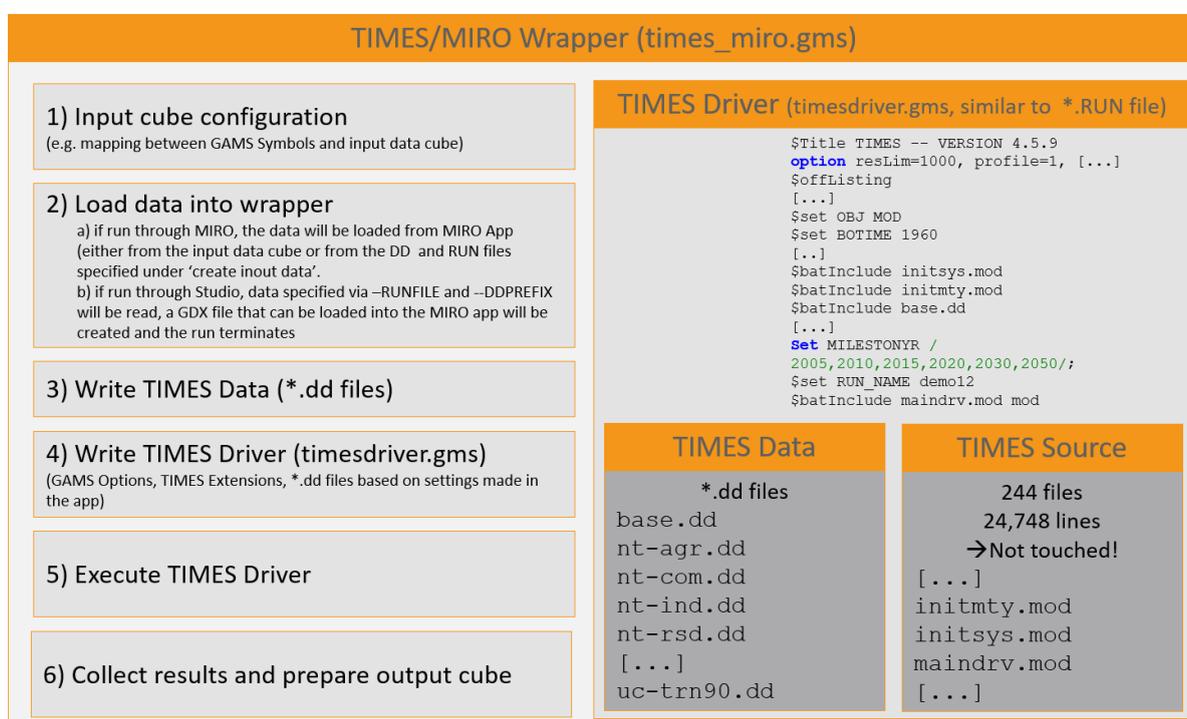
Thanks to the TIMES Cloud Service which is based on GAMS Engine, all three setups are already available for the TIMES/MIRO App.

- For the **MIRO Desktop** variant, [GAMS](#) and [GAMS MIRO](#) have to be installed locally and the user needs a professional GAMS license.
- For the **MIRO Desktop boosted by GAMS Engine** variant, only [GAMS MIRO](#) has to be installed locally. The user needs access rights (i.e. credentials and the URL) for the TIMES cloud Service (or any other GAMS Engine that she is entitled to use)
- For the **MIRO Server** variant, nothing has to be installed locally. The TIMES/App can be accessed through the web (<https://times.gams.com/>) with any common web browser. The user needs access rights for the TIMES cloud Service to which jobs will be submitted.

4. Implementation Details

The underlying technology of GAMS MIRO is based on R/Shiny, and it is already published under an open-source license on [GitHub](#). The source code and all configuration files of the TIMES/MIRO App are published under an open-source license on [GitHub](#).

The fundamental concept of the TIMES/MIRO App is that it builds on top of the well established TIMES concept that separates the TIMES source code from the data of a particular TIMES model which is provided via *.dd files and a *.run file. The heart of the TIMES/MIRO wrap is the [times_miro.gms](#) wrapper that combines GAMS and Python Code to implement the connection between TIMES and GAMS MIRO.



5. Documentation

The structure of the TIMES/MIRO App and its usage are documented in detail in the corresponding README files:

- https://github.com/GAMS-dev/TIMES_MIRO/blob/master/README.md
- https://github.com/GAMS-dev/TIMES_MIRO/blob/master/app_README.md