



GAMS MIRO

Introduction to a web interface for your GAMS models

Steve Dirkse & Lutz Westermann

GAMS Development Corp / GAMS Software GmbH



Model Stages

Design

Build

Deploy

GAMS: building models

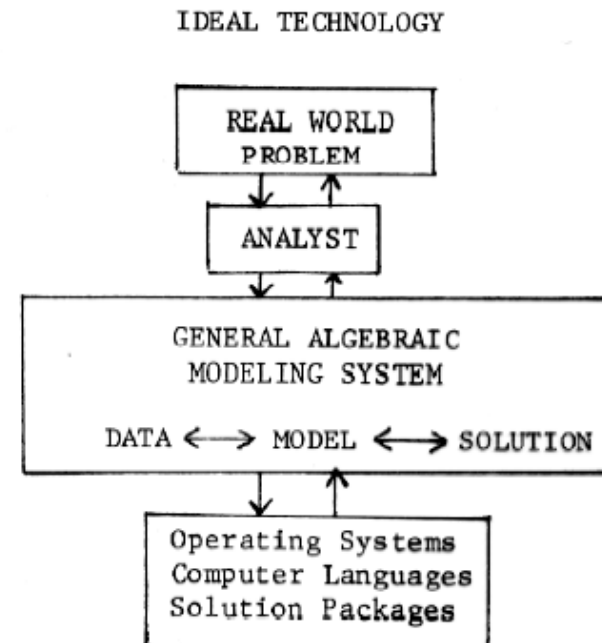
History

Design principles

1976 – A World Bank Slide

The
Vision

GAMS came into being!



- RESULT:
- Limited drain of resources
 - Same representation of models for humans and machines
 - Model representation is also model documentation

Simple Example

Indices:

i = plants

j = markets

Given Data:

a_i = supply of commodity of plant i (in cases)

b_j = demand for commodity at market j

c_{ij} = cost per unit shipment between plant i and market j

Decision Variables:

x_{ij} = amount of commodity to ship from plant i to market j

where $x_{ij} \geq 0$, for all i, j

Constraints:

Observe supply limit at plant i : $\sum_j x_{ij} \leq a_i$ for all i (cases)

Satisfy demand at market j : $\sum_i x_{ij} \geq b_j$ for all j (cases)

Objective Function: Minimize $\sum_i \sum_j c_{ij} x_{ij}$ (\$K)

Sets

```
i  canning plants  / seattle, san-diego /  
j  markets          / new-york, chicago, topeka / ;
```

Parameters

```
a(i)  capacity of plant i in cases  
b(j)  demand at market j in cases  
c(i,j) transport cost in thousands of dollars per case ;
```

Variables

```
x(i,j) shipment quantities in cases  
z      total transportation costs in thousands of dollars ;
```

Equations

```
cost      define objective function  
supply(i) observe supply limit at plant i  
demand(j) satisfy demand at market j ;
```

```
cost ..      z  =e=  sum((i,j), c(i,j)*x(i,j)) ;  
supply(i) ..  sum(j, x(i,j))  =l=  a(i) ;  
demand(j) ..  sum(i, x(i,j))  =g=  b(j) ;
```

```
Model transport /all/ ;
```

What did this give us?

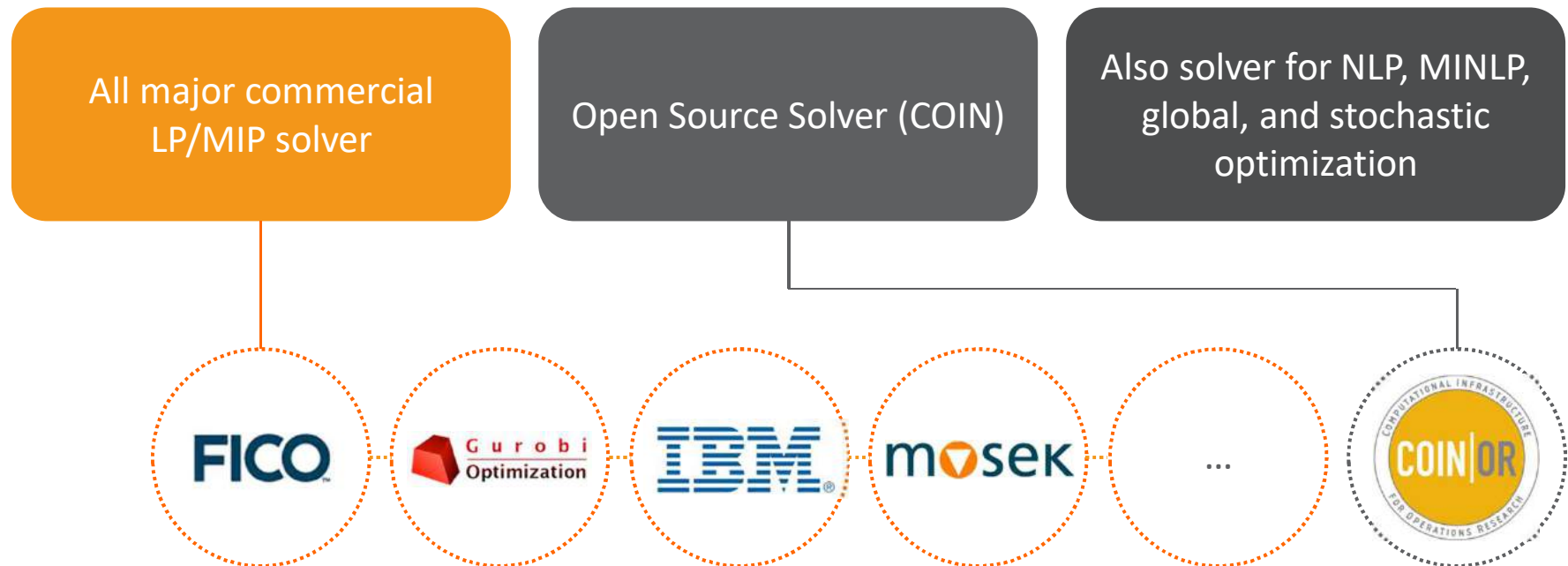
- Simplified model development & maintenance
- Increased productivity tremendously
- Made mathematical optimization available to a broader audience (domain experts)

2012 INFORMS Impact Prize:

“The awardees were trailblazers in developing the five most important algebraic modeling languages: AIMMS, AMPL, GAMS, LINDO/LINGO and MPL. ...”

Independence of Model and Solver

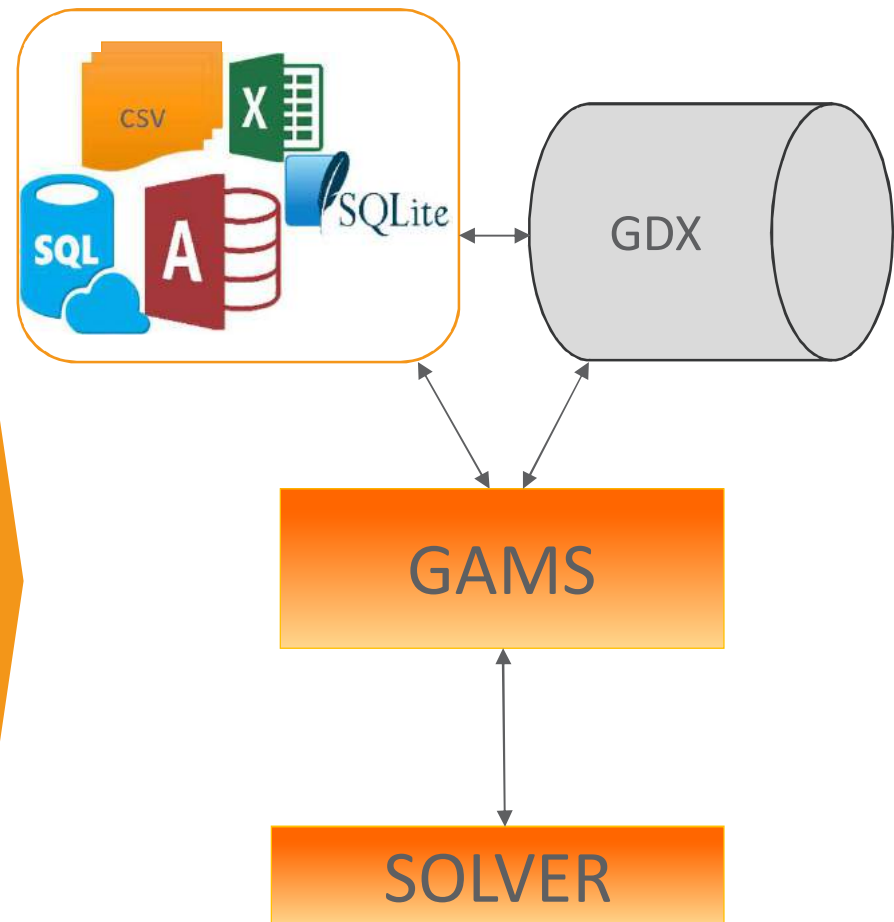
One environment for a wide range of model types and solvers



➔ Switching between solvers with one line of code!

Independence of Model and Data

- Declarative Modeling
- ASCII: Initial model development
- GDX: Data layer (“contract”) between GAMS and applications
 - Platform independent
 - No license required
 - Direct GDX interfaces and general API
 - ...



Independence of Model and User Interface

API's

- *Low Level*
- **Object Oriented:** .Net, Java, Python, C++
- No modeling capability:
Model is written in GAMS
- Wrapper class that
encapsulates a GAMS model



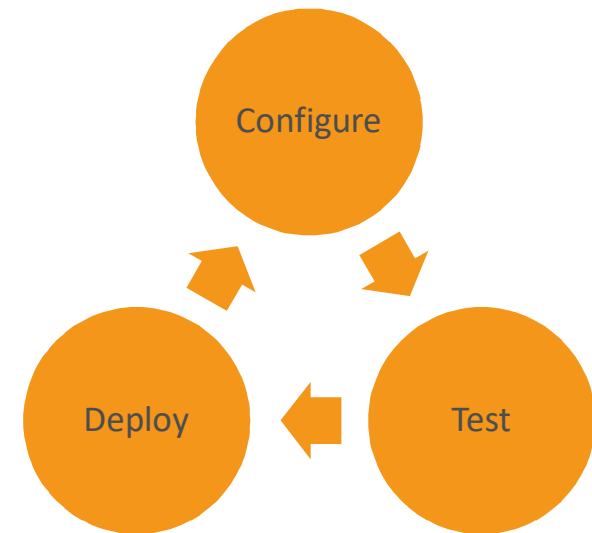
MIRO backstory

Client request for a web deployment environment
Benefits of R/Shiny

Motivation

Separation of Tasks:

- Modeling work
 - AMLs are powerful tools for developing solver-independent optimization models
 - GAMS for modeling and optimization
 - Intuitive deployment and visualization are becoming increasingly important
 - End-users of optimization software are very often not modeling experts
 - Need for easy-to-use tool to visualize data and compare results
- Current deployment possibilities are not satisfactory for everyone

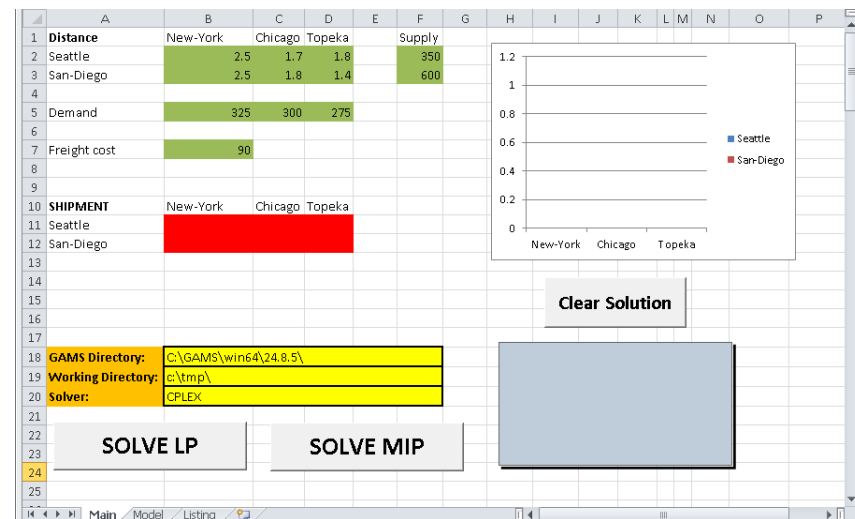


Deployment of GAMS models

current possibilities

Expert level APIs

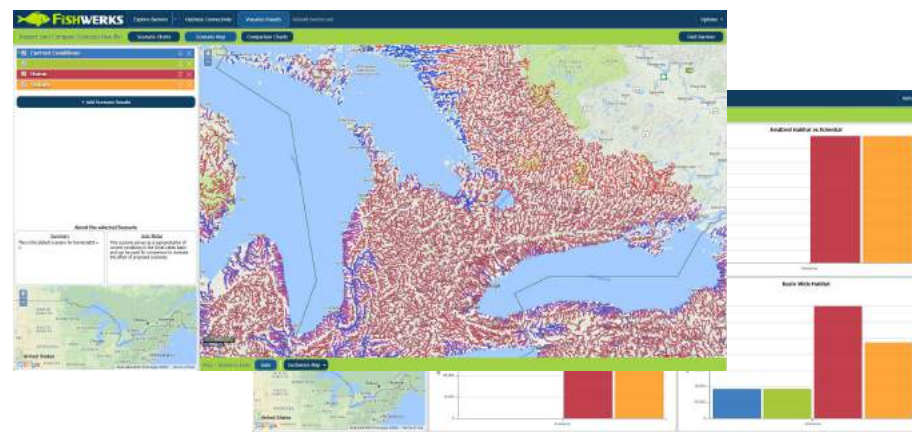
- GDX, OPT, GAMSX, GMO, ...
- High performance and flexibility
- Automatically generated imperative APIs for several languages (C, C++, C#, Delphi, Java, Python, VBA, ...)



Object Oriented APIs

- GAMS comes with several OO APIs (Python, Java, C++, C#, ...) to develop applications

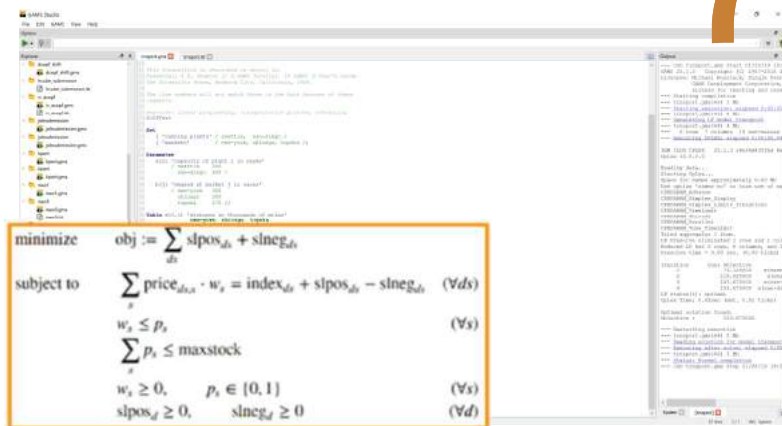
→ Programming required to build your applications



GAMS MIRO

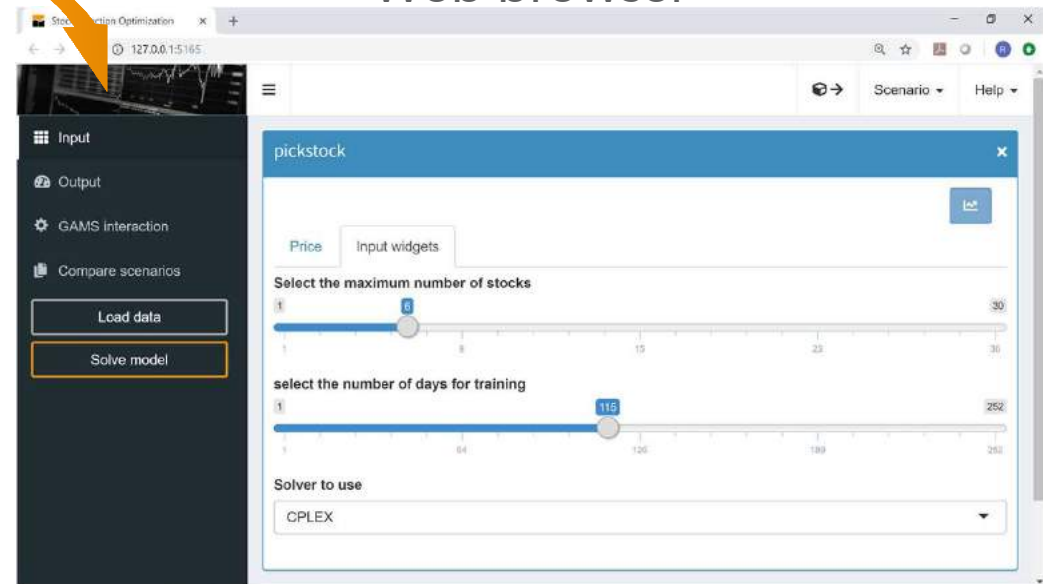
Model Interface with Rapid Orchestration

Develop GAMS model

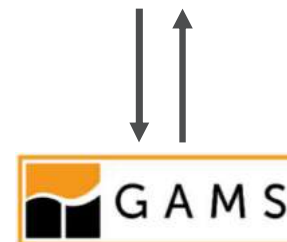


Click to
deploy

Web browser



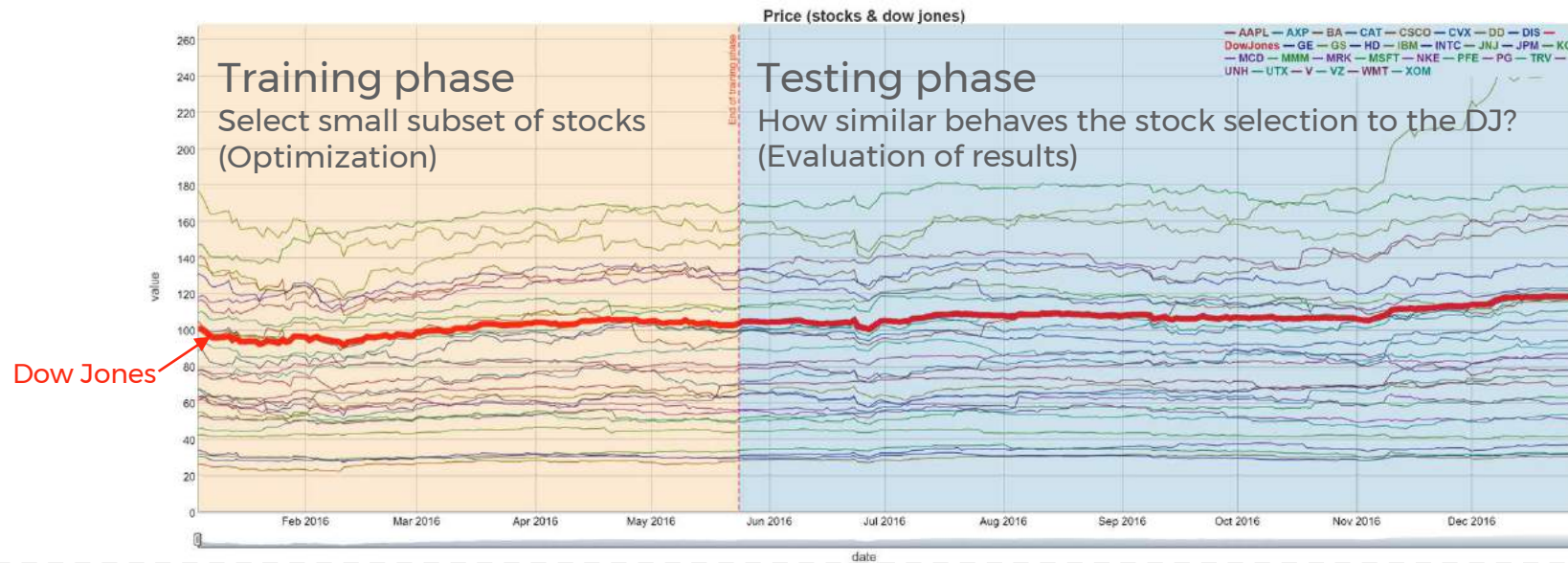
- ✓ Web interface for GAMS models
- ✓ Usage via web browser
- ✓ GAMS as a black box
- ✓ Focus on automated deployment
- ✓ Configuration instead of programming



Example

Model: Pickstock

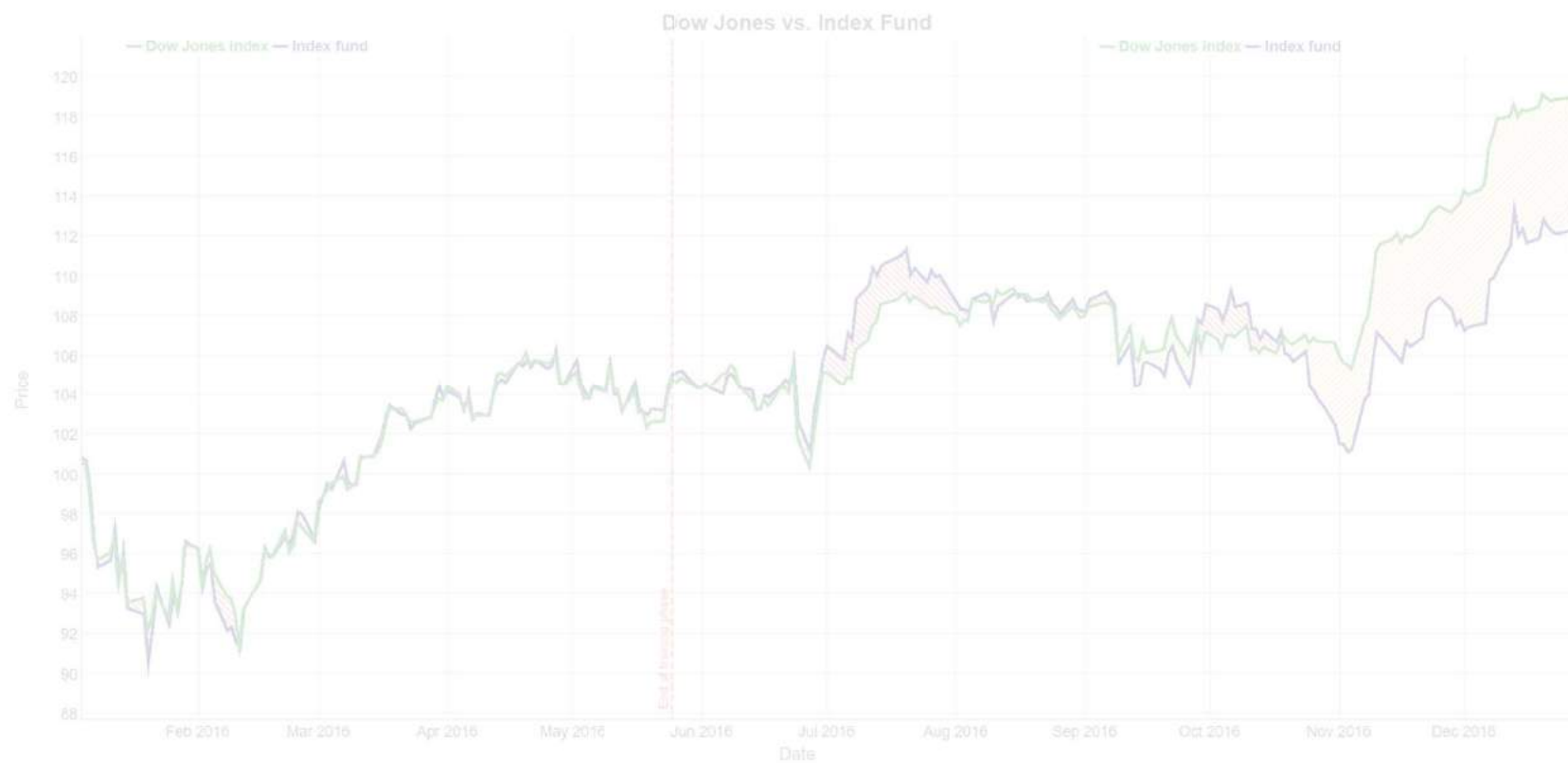
Model: *Pickstock*



- **Data:** Performance of all shares of the Dow Jones index over a period of 1 year
- **Goal:** Find a small selection of stocks that follows the Dow Jones as good as possible
- **Optimization model:** Select a subset ($\leq \text{maxstock}$) of Dow Jones stocks, along with weights, so that this portfolio behaves similarly to the overall index (in the training phase)

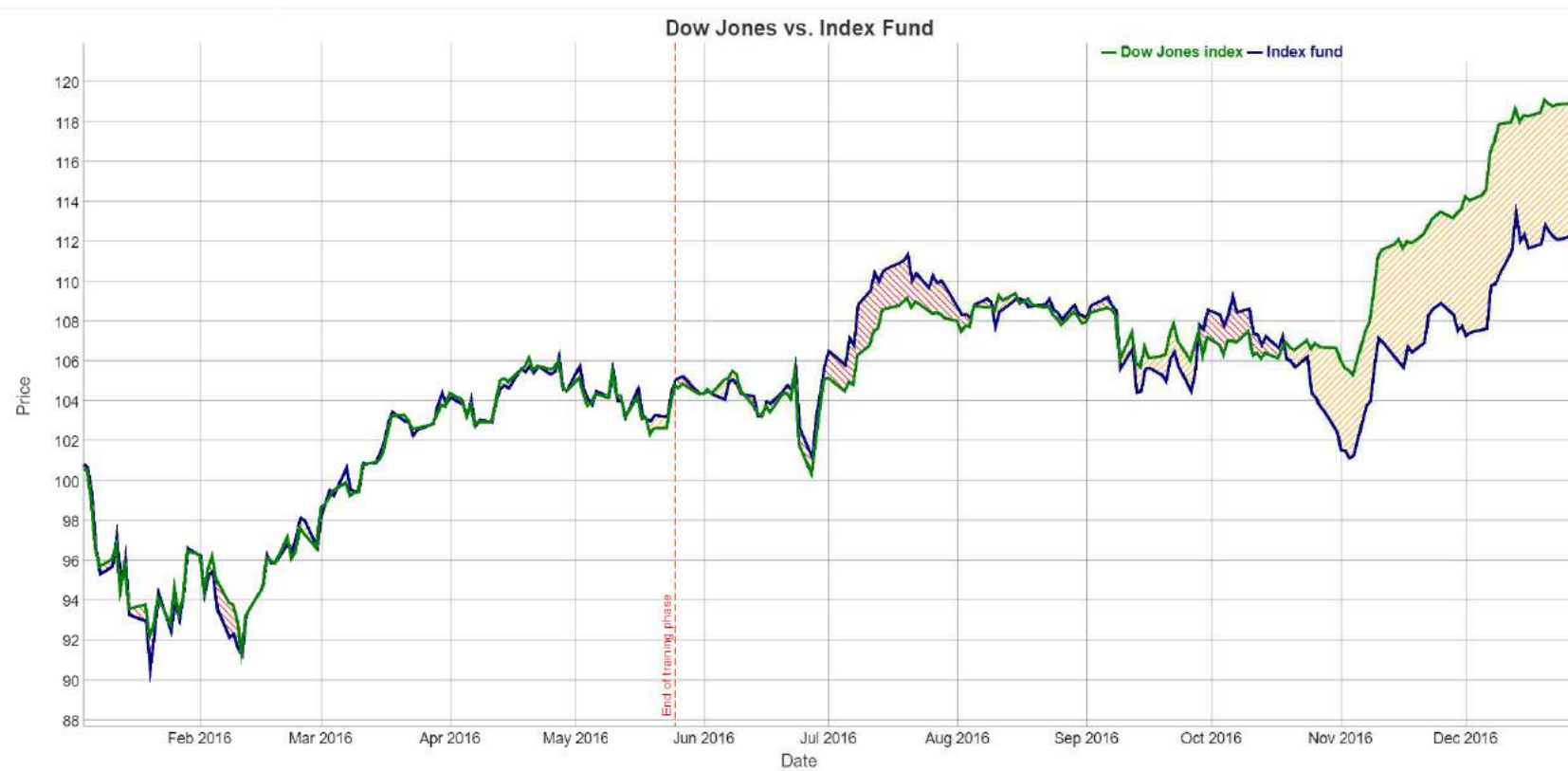
$$\begin{aligned}
 &\text{minimize} && \text{obj} := \sum_{ds} \text{slpos}_{ds} + \text{slneg}_{ds} \\
 &\text{subject to} && \sum_s \text{price}_{ds,s} \cdot w_s = \text{index}_{ds} + \text{slpos}_{ds} - \text{slneg}_{ds} \quad (\forall ds) \\
 & && w_s \leq p_s \quad (\forall s) \\
 & && \sum_s p_s \leq \text{maxstock} \\
 & && w_s \geq 0, \quad p_s \in \{0, 1\} \quad (\forall s) \\
 & && \text{slpos}_d \geq 0, \quad \text{slneg}_d \geq 0 \quad (\forall d)
 \end{aligned}$$

Model: *Pickstock*



$$\text{minimize} \quad \text{obj} := \sum_{ds} \text{slpos}_{ds} + \text{slneg}_{ds}$$

Model: *Pickstock*



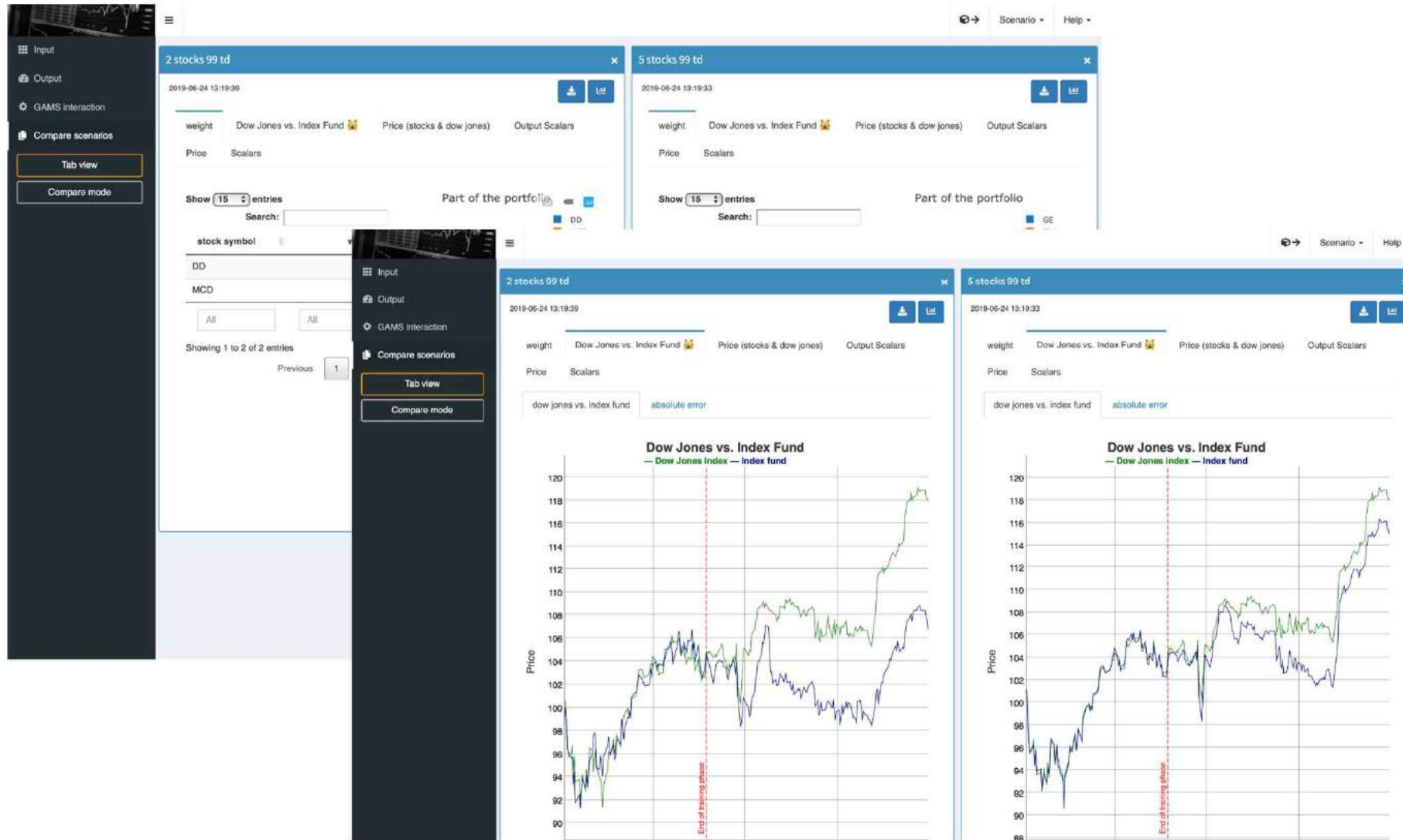
$$\text{minimize} \quad \text{obj} := \sum_{ds} \text{slpos}_{ds} + \text{slneg}_{ds}$$

Demo: Run

Look at an app built with MIRO



Compare scenarios



Demo: How-to

Step by step:

Deployment of a GAMS model with GAMS MIRO

GAMS MIRO || DATA CONCEPT

GAMS model

```

Set
  i 'canning plants' / seattle, san-diego /
  j 'markets' / new-york, chicago, topeka /;

Parameter
  a(i) 'capacity of plant i in cases'
    / seattle 350
      san-diego 600 /;

$onExternalInput
  b(j) 'demand at market j in cases'
    / new-york 325
      chicago 300
      topeka 275 /;

Table d(i,j) 'distance in thousands of miles'
  / new-york chicago topeka
seattle 2.5 1.7 1.8
san-diego 2.5 1.8 1.4 /;

Scalar f 'freight in dollars per case per thousand miles' / 90 /;
$offExternalInput

Parameter c(i,j) 'transport cost in thousands of dollars per case';
c(i,j) = f*d(i,j)/1000;

Variable
  x(i,j) 'shipment quantities in cases'
  z 'total transportation costs in thousands of dollars';

Positive Variable x;

Equation
  cost 'define objective function'
  supply(i) 'observe supply limit at plant i'
  demand(j) 'satisfy demand at market j';

cost.. z =e= sum((i,j), c(i,j)*x(i,j));
supply(i).. sum(j, x(i,j)) =l= a(i);
demand(j).. sum(i, x(i,j)) =g= b(j);

Model transport / all /;
solve transport using lp minimizing z;

*reporting
Set scheduleHdr 'schedule header' / 'cap', 'demand', 'quantities' /;

$onExternalOutput
  Parameter
    schedule(i,j,scheduleHdr) 'shipment quantities in cases'
    total_cost 'total transportation costs in thousands of dollars';
$offExternalOutput

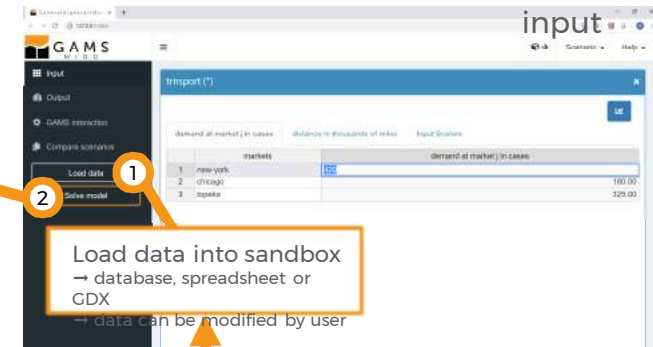
total_cost = z.l;
schedule(i,j,'cap') = a(i);
schedule(i,j,'demand') = b(j);
schedule(i,j,'quantities') = x.l(i,j);
  
```

MIRO independent

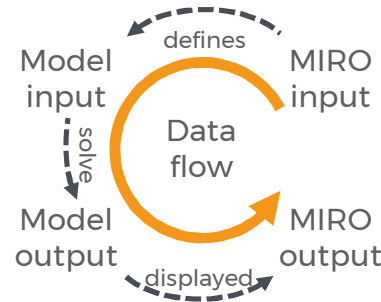
MIRO independent

Data in model overwritten when MIRO is used

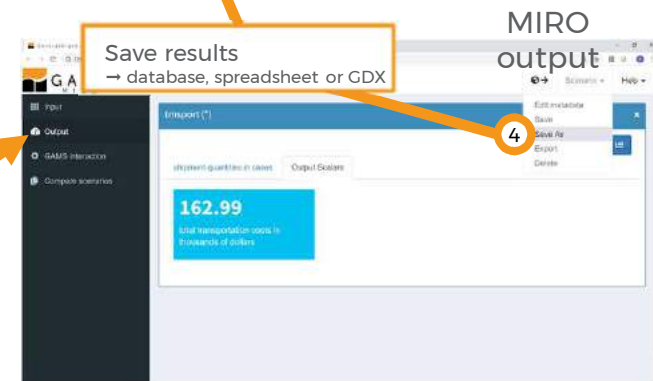
Solve model
MIRO input data used for model run



Load data into sandbox
→ database, spreadsheet or GDX
→ data can be modified by user



Save results
→ database, spreadsheet or GDX



Return results
Tagged model output visualized in MIRO

Data exchange between GAMS model and MIRO

- 1 Data is loaded into the MIRO interface. These come either from the internal database or from external data sources such as Excel or GDX, but NOT directly from the GAMS model.
- 2 When clicking on solve, the data visible in the MIRO input section is sent to the GAMS model. All data of affected symbols in the model will be overwritten. Symbols that are not visible in MIRO are not touched.
- 3 The model is now run with the updated data. Afterwards the results are sent to MIRO and displayed in the output section.
- 4 From there, data can be saved in the internal database and/or exported as a file.

Scenario runs

The GAMS MIRO Hypercube mode

Hypercube mode

- Generate and analyze multiple scenarios in batch mode
- Goals:
 - Little to no effort to create interface
 - Generate only scenarios that have not yet been solved
 - Scenario management should be organized as clearly as possible
 - E.g. apply sophisticated filters to see only scenarios fulfilling certain criteria
 - Perform analyses related to KPIs and/or other indicators

Hypercube mode

scenario generation

Base mode

Hypercube mode

The image shows two side-by-side screenshots of the 'pickstock' application interface. The top window, labeled 'Base mode', has a blue header with the title 'pickstock' and a close button. It features two tabs: 'Price' (selected) and 'Input widgets'. Below the tabs, there are two sliders: 'Select the maximum number of stocks' with a value of 6, and 'select the number of days for training' with a value of 115. A 'Solver to use' dropdown menu is set to 'CPLEX'. The bottom window, labeled 'Hypercube mode', also has a blue header with the title 'pickstock' and a close button. It has the same 'Price' and 'Input widgets' tabs. The 'Select the maximum number of stocks' slider has a value of 13, and the 'select the number of days for training' slider has a value of 140. To the right of these sliders are 'Step size' input fields with values 1 and 5. The 'Solver to use' dropdown menu is set to 'CPLEX' and includes buttons for 'CPLEX', 'XPRESS', and 'CBC'. A large grey arrow points from the 'Base mode' window to the 'Hypercube mode' window, indicating a transition or comparison.

pickstock

Price Input widgets

Select the maximum number of stocks

1 6 30

select the number of days for training

1 115 12

Solver to use

CPLEX

pickstock

Price Input widgets

Select the maximum number of stocks

1 2 13 30

Step size

1

select the number of days for training

1 35 140 252

Step size

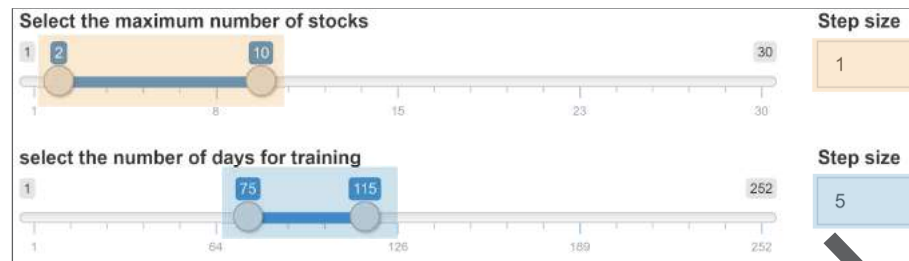
5

Solver to use

CPLEX XPRESS CBC

Hypercube mode

scenario generation



+ 1

	2	3	4	5	6	7	8	9	10
75	1	2	3	4	5	6	7	8	9
80	10	11	12	13	14	15	16	17	18
85	19	20	21	22	23	24	25	26	27
90	28	29	30	31	32	33	34	35	36
95	37	38	39	40	41	42	43	44	45
100	46	47	48	49	50	51	52	53	54
105	55	56	57	58	59	60	61	62	63
110	64	65	66	67	68	69	70	71	72
115	73	74	75	76	77	78	79	80	81

+ 5

Number of scenarios
= Cartesian product of scalar input combinations

Hypercube mode

Data import & monitoring of scenario runs

Import data

Owner	Submission date	Job tags	Status	Action
user	2019-01-18 15:05:09	runxy	running	Show log Discard
user	2019-01-18 15:04:53		running	Show log Discard
user	2019-01-18 15:04:42	run1	completed	Import Show log Discard
user	2019-01-18 15:02:57	all_types	completed	Import Show log Discard
user	2019-01-18 14:58:50	MIP min_ship	completed	Import Show log Discard

Show history

Manual import

Hypercube mode

Scenario management

Load scenarios

Time created

between

2019-02-01

to

2019-02-21

-

Job tags

is

superman

-

AND

Time created

between

2019-02-22

to

2019-03-04

-

Job tags

is

wonder woman

-

AND

OR

Fetch results

Show

10

entries

Search:

Owner	Time created	Job tags	maximum number of stocks to select	MIP-Solver	number of days for training	Ratio between error test and error train	Absolute error in entire testing phase	Absolute error in entire training phase	last date of training period
user	2/15/2019, 2:53:55 PM	superman	8	CPLEX	99	13.7001401311091	170.448122203935	12.4413415171496	2016-05-24
user	2/15/2019, 2:53:55 PM	superman	3	CPLEX	99	5.13673004574033	229.061239845358	44.592812510229	2016-05-24
user	2/15/2019, 2:53:55 PM	superman	24	CPLEX	99	19.3422042295776	15.0432531197135	0.777742440373459	2016-05-24

All

All

All

All

All

All

Showing 1 to 10 of 220 entries

Choose selected scenarios

Choose current page

Choose all

✓ Apply sophisticated filters to get only scenarios fulfilling certain criteria

Hypercube mode

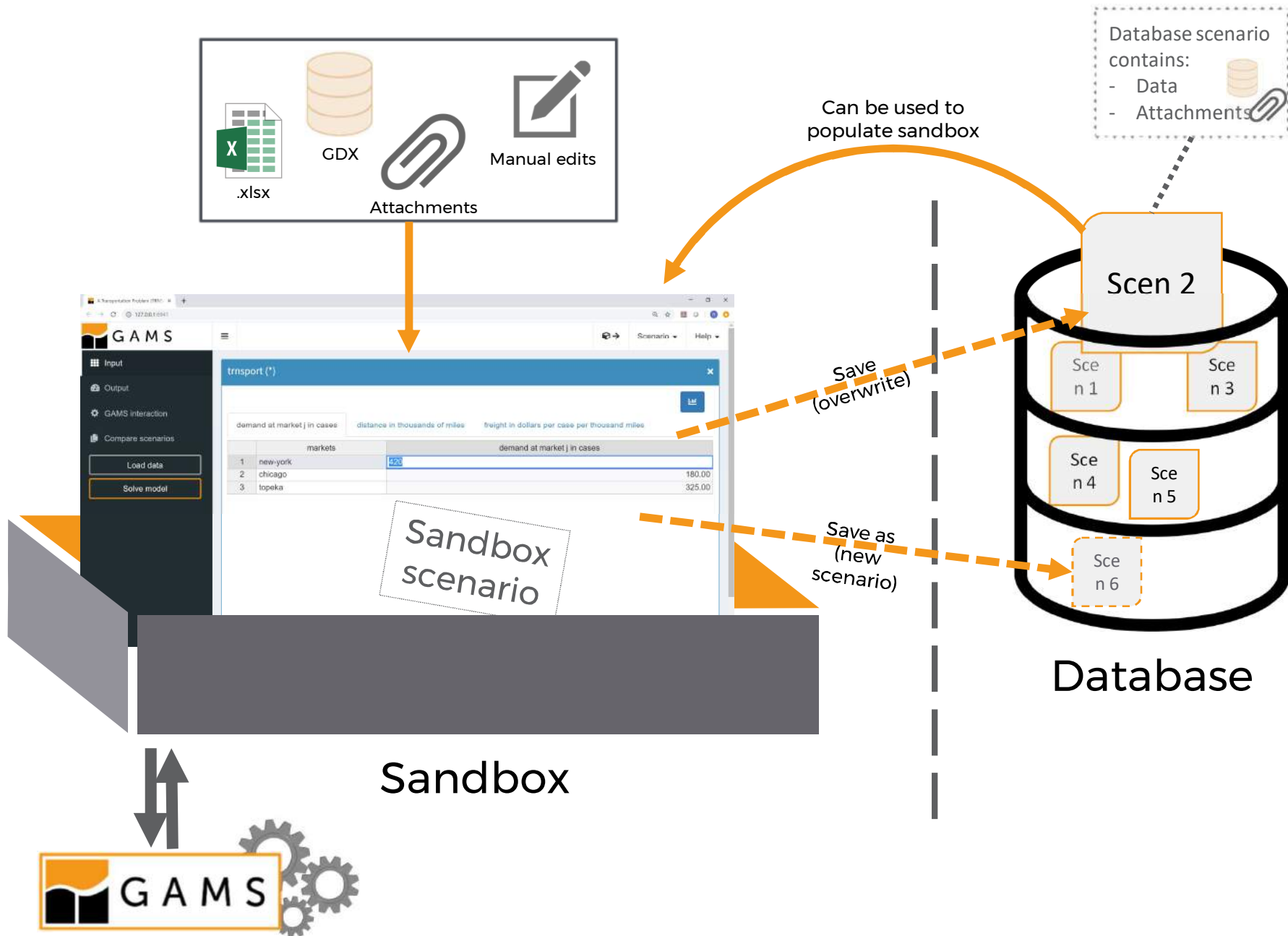
Analysis



Demo

Hypercube mode

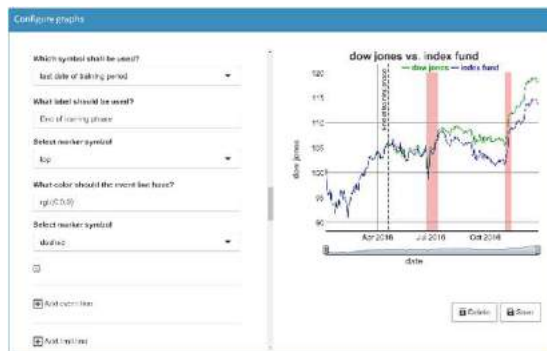
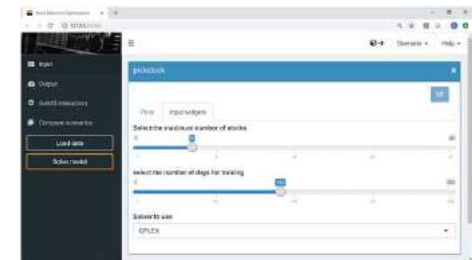
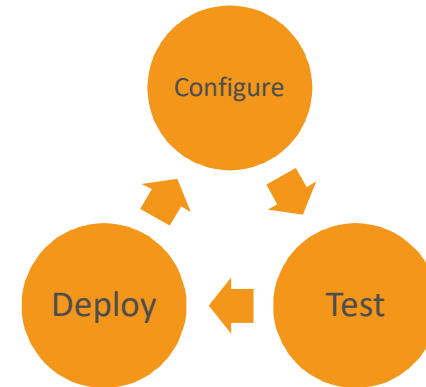
GAMS MIRO || SCENARIO CONCEPT



Summary

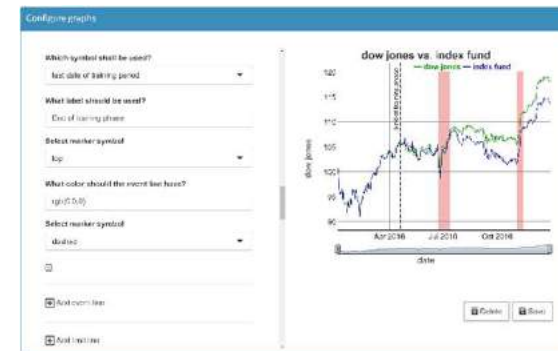
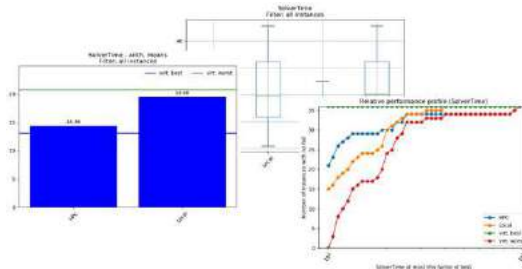
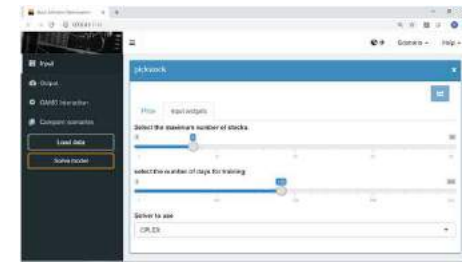
Summary

- Separation of tasks:
 - Modeling work
 - Model deployment
 - In OR projects often over several iterations
 - model deployment should not take much time (at least during these iterations)
- End-users are very often not modeling experts
- Quick & automated deployment of GAMS models
- Data visualization with charts / graphics
- Easy to configure
- Intuitive and structured work without GAMS knowledge
- Easy and convenient data and scenario management



Summary

- Quick & automated deployment of GAMS models
- Data visualization with powerful charts / graphics
- Configuration mode
- Generation, processing and evaluation of scenario data
- Generation of performance statistics and sensitivity analyses
- Data export for external GAMS jobs and analyses
- Intuitive and structured work without GAMS knowledge
- Easy and convenient data and scenario management



Outlook

MIRO 1.0

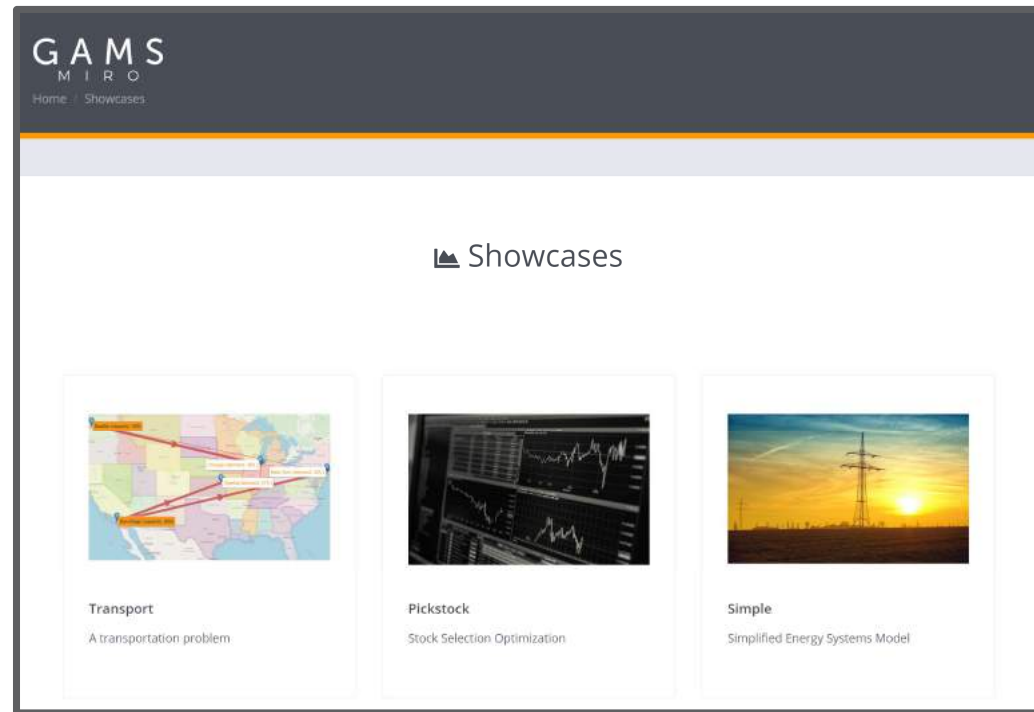
Outlook

MIRO 1.0

- Showcases

Already online!

<https://www.gams.com/miro/index.html>



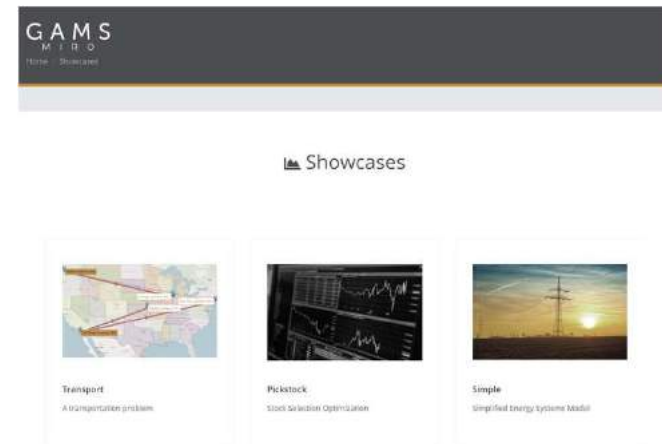
You want to share your own MIRO app with others?

In order to show you the different application possibilities of MIRO, we want to expand our gallery with MIRO showcase apps permanently. If you have configured a MIRO app and would like to share it with others, you are welcome to do so here! Contact miro@gams.com if you want to share your app with us.

Outlook

MIRO 1.0

- Showcases
- Data validation – MIRO log



GAMS output

Log file
Listing file
MIRO log file

```
-----
Executing transport
-----
a:: Capacity insufficient to meet demand
```

capacity of plant i in cases

demand at market j in cases

Market location information

Input widgets

• **Capacity insufficient to meet demand**

	canning plants	capacity of plant i in cases
1	Seattle	200.00
2	San-Diego	600.00

Outlook

MIRO 1.0

- Showcases
- Data validation – MIRO log
- Custom output for more detailed analyses (e.g. Jupyter notebooks)

GAMS output

Log file
Listing file
MIRO log file

Executing transport

a:: Capacity insufficient to meet demand

capacity of plant i in cases demand at i

Market location information Input widget

• Capacity insufficient to meet demand

	canning plants	capacity of plant
1	Seattle	200.00
2	San-Diego	600.00



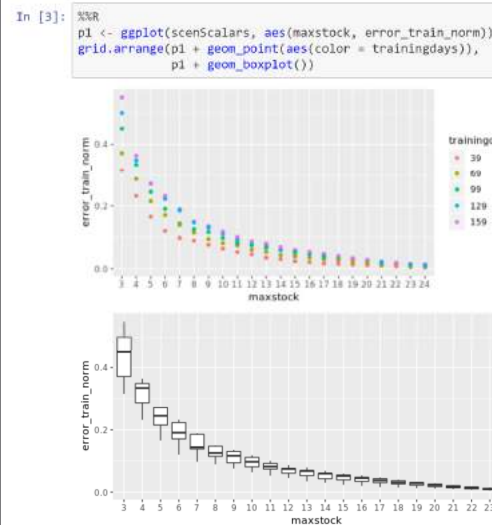
Showcases



```
col_types = cols()[c(1,3)] %>%
  spread(scalar, value, convert = TRUE)
)) %>%
  mutate(error_train_norm = error_train/trainingdays,
         error_test_norm = error_test/(252 - trainingdays),
         error_ratio_norm = error_ratio/trainingdays) %>%
  mutate(maxstock = as.factor(maxstock),
         trainingdays = as.factor(trainingdays))
kable(head(scenScalars), caption = "Raw scalar data")
```

error_ratio	error_test	error_train	lastdaytraining	maxstock	mip	trainingdays	error_train_norm	error_test_norm	err
27.894663	26.11184	0.9360871	2016-05-24	23	SCIP	99	0.0094554	0.1706656	
12.118973	26.06389	2.1506685	2016-07-07	22	SCIP	129	0.0166718	0.2119816	
7.109752	249.03047	35.0266060	2016-07-07	5	SCIP	129	0.2715241	2.0246380	
172.617428	63.00389	0.3649915	2016-02-29	21	SCIP	39	0.0093588	0.2957929	
4.166327	125.21678	30.0544814	2016-08-18	7	SCIP	159	0.1890219	1.3464170	
8.283518	119.10166	14.3781494	2016-07-07	10	SCIP	129	0.1114585	0.9683061	

Now that we have the data in the shape we want, we can plot it. Let's first look at the training error:



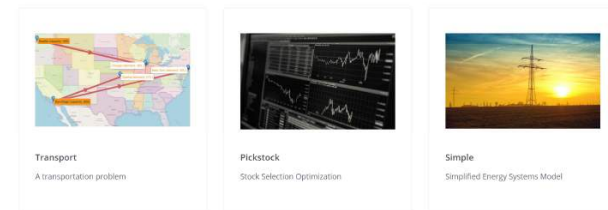
Outlook

MIRO 1.0

- Showcases
- Data validation – MIRO log
- Custom output for more detailed analyses (e.g. Jupyter notebooks)
- Graphs as data input



Showcases



GAMS output

Log file Listing file MIRO log file

Executing transport

a:: Capacity insufficient to meet demand

capacity of plant i in cases

demand at market j in cases

Market location information

Input widgets

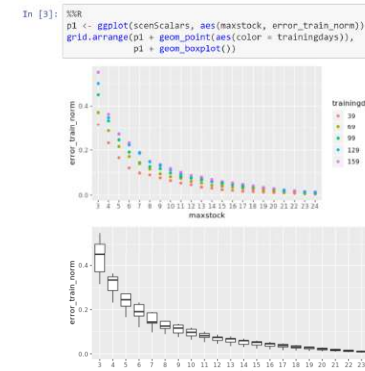
• Capacity insufficient to meet demand

	canning plants	capacity of plant i in cases
1	Seattle	200.00
2	San-Diego	600.00

```
col_types = cols()))[c(1,3)] %>%
spread(scalar, value, convert = TRUE)
))) %>%
mutate(error_train_norm = error_train/trainingdays,
error_test_norm = error_test/(352 - trainingdays),
error_ratio_norm = error_ratio/trainingdays) %>%
mutate(maxstock = as.factor(maxstock),
trainingdays = as.factor(trainingdays))
kable(head(scenScalars), caption = "Raw scalar data")
```

error_ratio	error_test	error_train	lastdaytraining	maxstock	inip	trainingdays	error_train_norm	error_test_norm	err
27.894663	26.11184	0.9360872	2016-05-24	23	SCIP	99	0.0094554	0.1706566	
12.118973	26.06389	2.1506685	2016-07-07	22	SCIP	129	0.0166718	0.2119016	
7.189752	249.03847	35.0266068	2016-07-07	5	SCIP	129	0.2715241	2.0246388	
172.617428	63.00389	6.3649915	2016-02-29	21	SCIP	139	0.0093588	0.2957929	
4.166327	125.21678	30.0544814	2016-08-18	7	SCIP	159	0.1890219	1.3464170	
8.283518	119.18166	14.3781494	2016-07-07	10	SCIP	129	0.1114585	0.9683861	

Now that we have the data in the shape we want, we can plot it. Let's first look at the training error:



Outlook

MIRO 1.0

The screenshot displays the Outlook MIRO 1.0 web application interface. The browser window title is "Stock Selection Optimization". The address bar shows the URL "127.0.0.1:6731". The application has a dark theme with a teal accent color. On the left, a sidebar contains navigation links: "Input", "Output", "GAMS Interaction", and "Compare scenarios". Below these are two buttons: "Load data" and "Solve model". The main content area is titled "default (*)" and features two tabs: "Price" and "Input widgets". The "Input widgets" tab is active, showing two sliders and a dropdown menu. The first slider is labeled "select the number of days for training" and has a value of 175. The second slider is labeled "Solver to use" and has a value of CPLEX. The "Solve model" button is highlighted with an orange border.

Stock Selection Optimization

127.0.0.1:6731

Login Scenario Help

Input

Output

GAMS Interaction

Compare scenarios

Load data

Solve model

default (*)

Price Input widgets

select the number of days for training

1 30

1 252

1 64 126 189 252

Solver to use

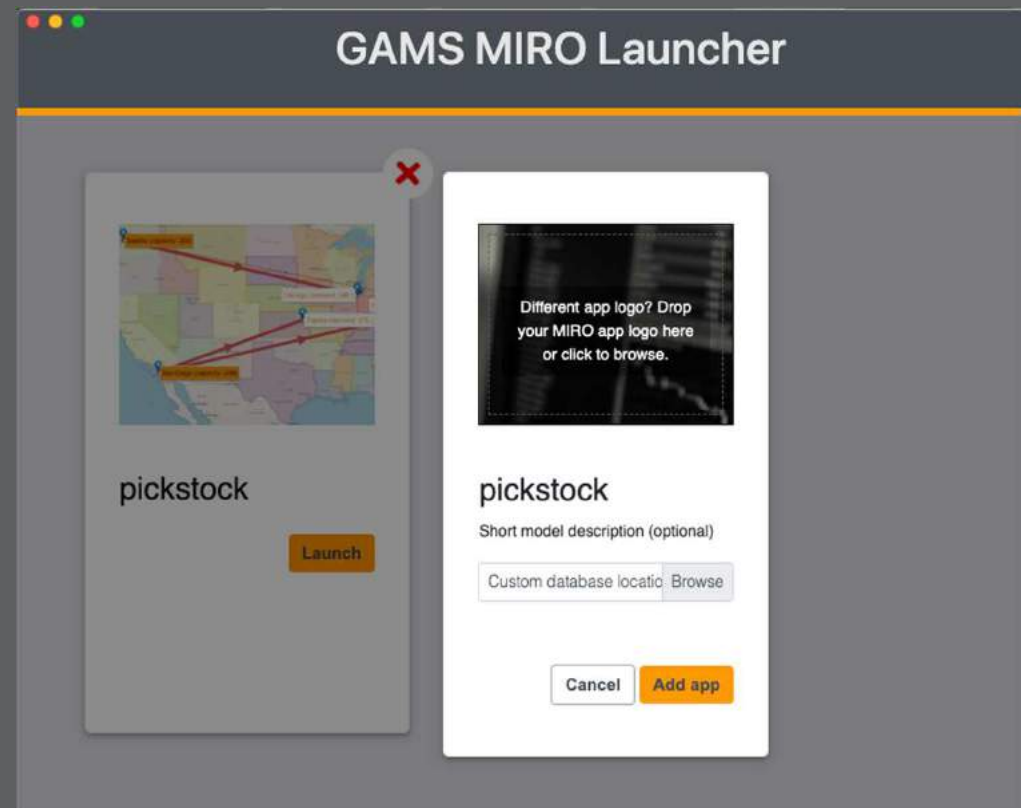
CPLEX

Outlook

MIRO 1.0

Deployment

- Select “Deploy App” from GAMS Studio
 - → Creates a MIRO App
- Add one or many of these Apps and add them to the MIRO Launcher
- Launcher allows to start App in browser or stand alone



Outlook

MIRO 1.0

Various other improvements done already:

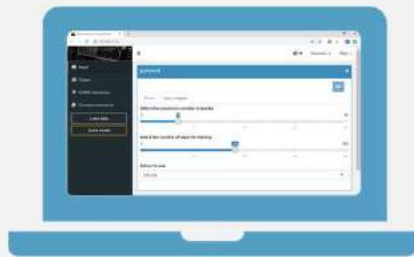
- Full Chinese language support
- Input/output symbols
- Heatmaps for input tables
- "Stop" with soft kill first, then hard kill
- Number of tabs visible can be configured (rest available via dropdown)
- Scenario comparison:
 - Input/output symbol tabs in different colors
 - Sandbox scenarios can be loaded directly
- Configuration mode restructured
- ...

Outlook: MIRO on a server

Enterprise features

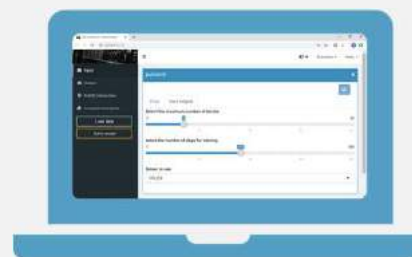
MIRO – job execution

Everything local

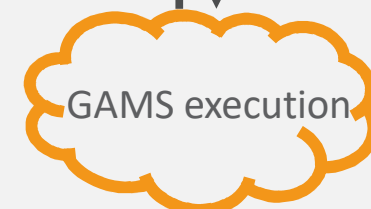


- GAMS installed
- MIRO installed
- Synchronous execution of GAMS jobs

Local MIRO application – remote GAMS execution



Everything on a server



Local MIRO – remote GAMS execution

- Only MIRO installed locally
- Computation on server
- Asynchronous execution of jobs

Security

- ✓ Authentication (for remote execution) handled by MIRO
- ✓ TLS enforced
- ✓ Certificate pinning
- ✓ Token instead of user password stored

Login required

Please enter your credentials below:

URL of the GAMS Remote Solver

Username

Password

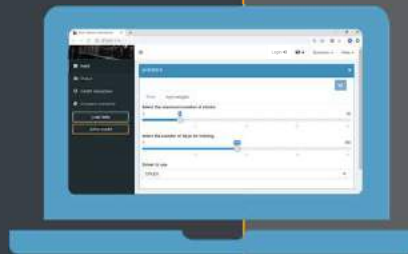
Namespace

Is your model registered in the namespace?

☐

Remember me

☒



Everything on a server

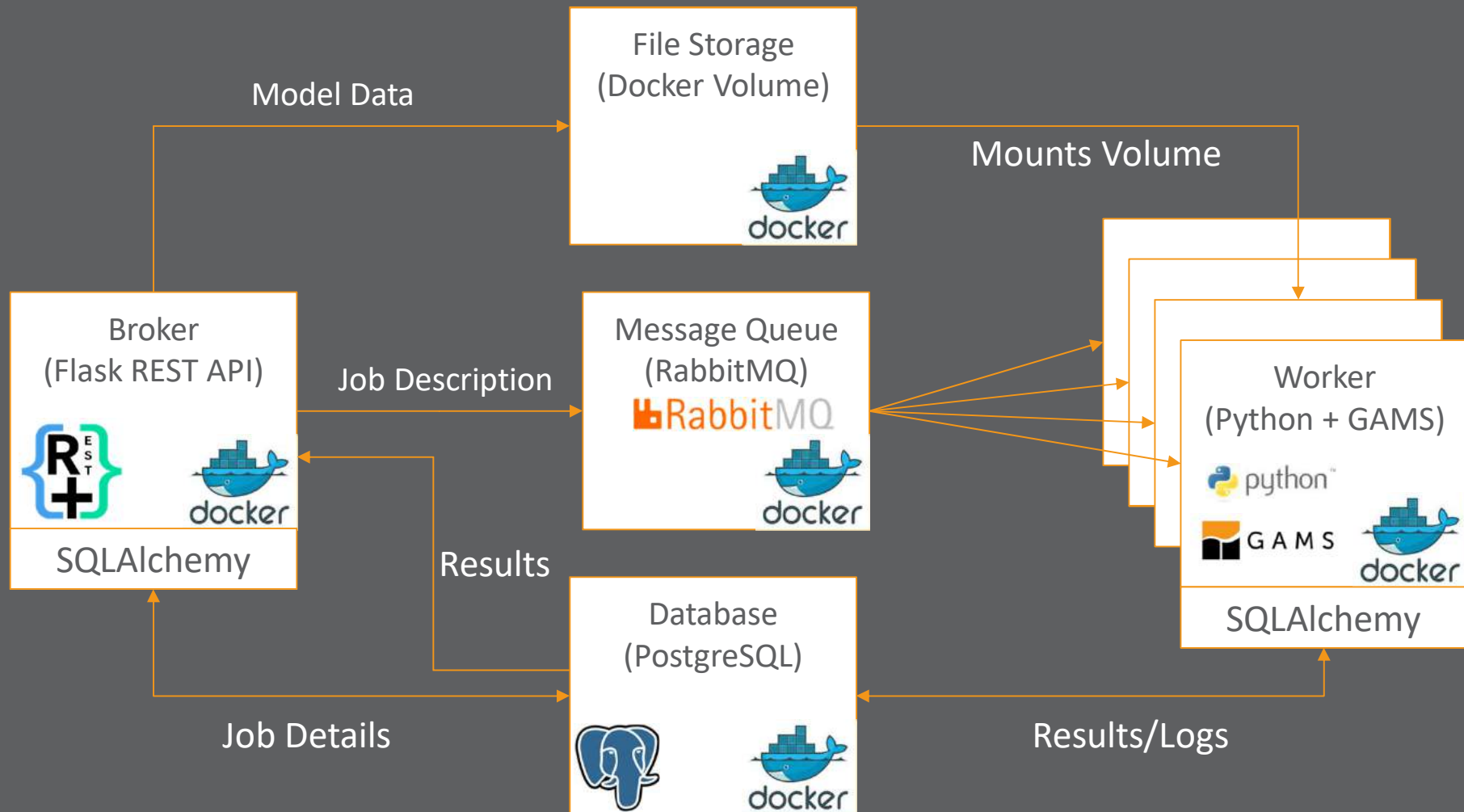
- Neither GAMS nor MIRO installation necessary
- Can be hosted by GAMS or on customer's server(s)

Security

- Authentication via LDAP, Keycloak, Kerberos, OpenID Connect, etc.
- Remote executor authentication disabled



Docker network



QUESTIONS?

Or: What do you miss?





For more information visit:
www.gams.com/miro

Meet us at the GAMS booth!