Manual for the ETM’s API

On using the computational power embedded in the Energy Transition Model within an external application

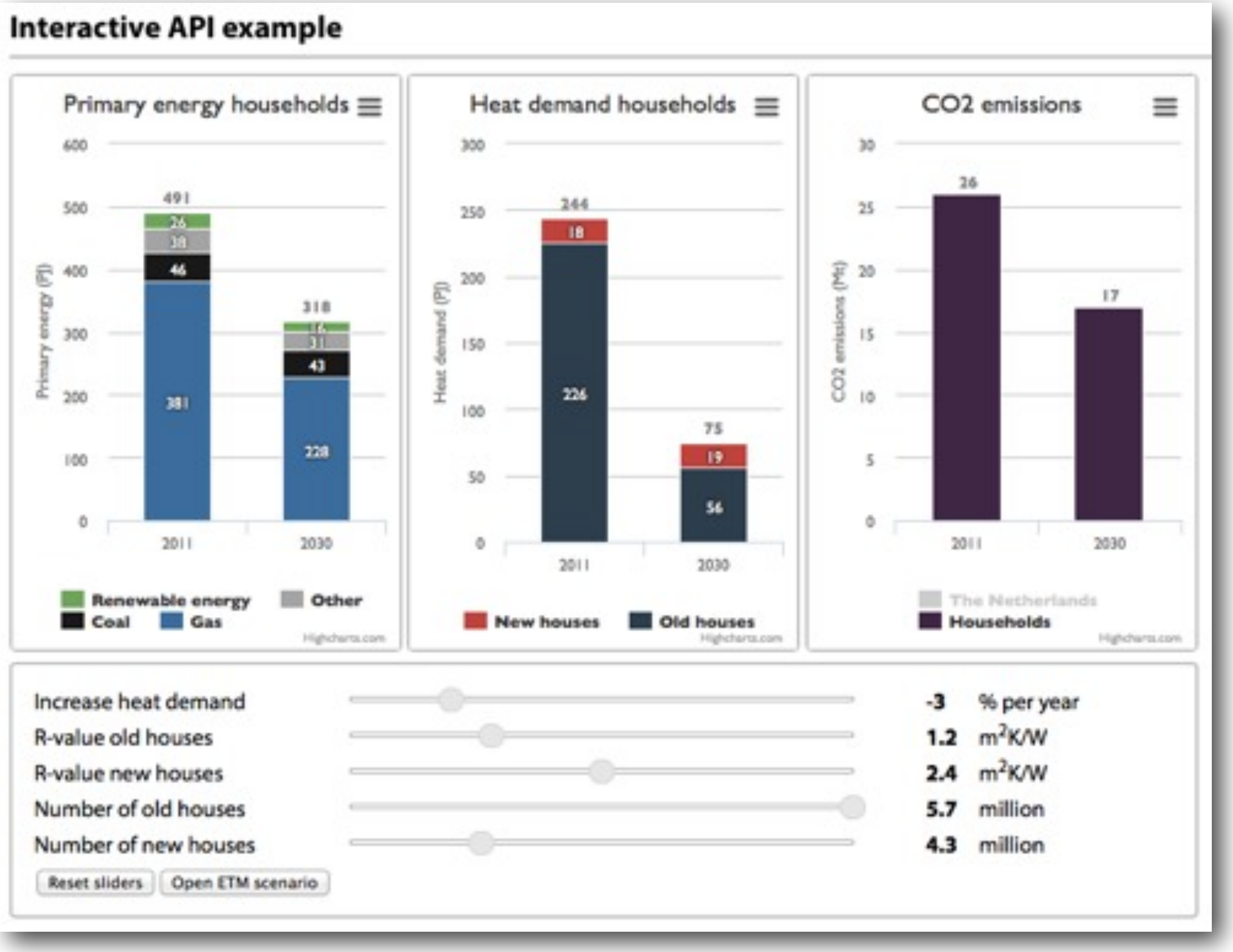
This live example shows how the ETM’s API can be used to focus on specific parts of the energy system, within an external application. All the code that is needed to run this example, can be found on this github repository.

The purpose of this manual is to show how the different parts of the example code interact.

The first five steps describe how the sliders and the graphics are initialized. The next six steps form a cycle that is repeated every time the user moves a slider.

It is convenient to have this manual side by side with the code, so you can see how the objects look like, which properties they have and what their methods do.

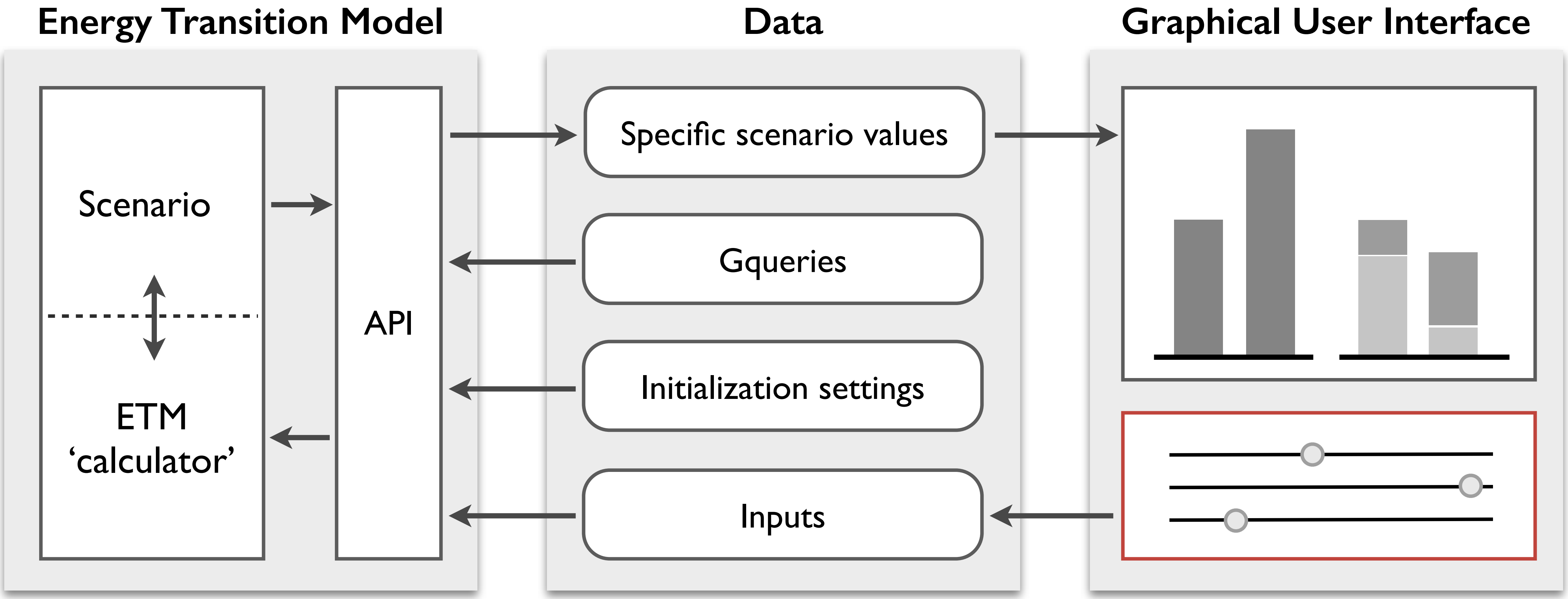
When running the live example, the different steps will be logged in the console. This way, you can easily indentify where in the code which processes are defined.



Interactive API example

Increase heat demand -3 % per year
R-value old houses 1.2 m^2K/W
R-value new houses 2.4 m^2K/W
Number of old houses 5.7 million
Number of new houses 4.3 million

If you run into any problems, don’t hesitate to send me an email at jonas.voorzanger@gmail.com, or give a call at 06-81958192.

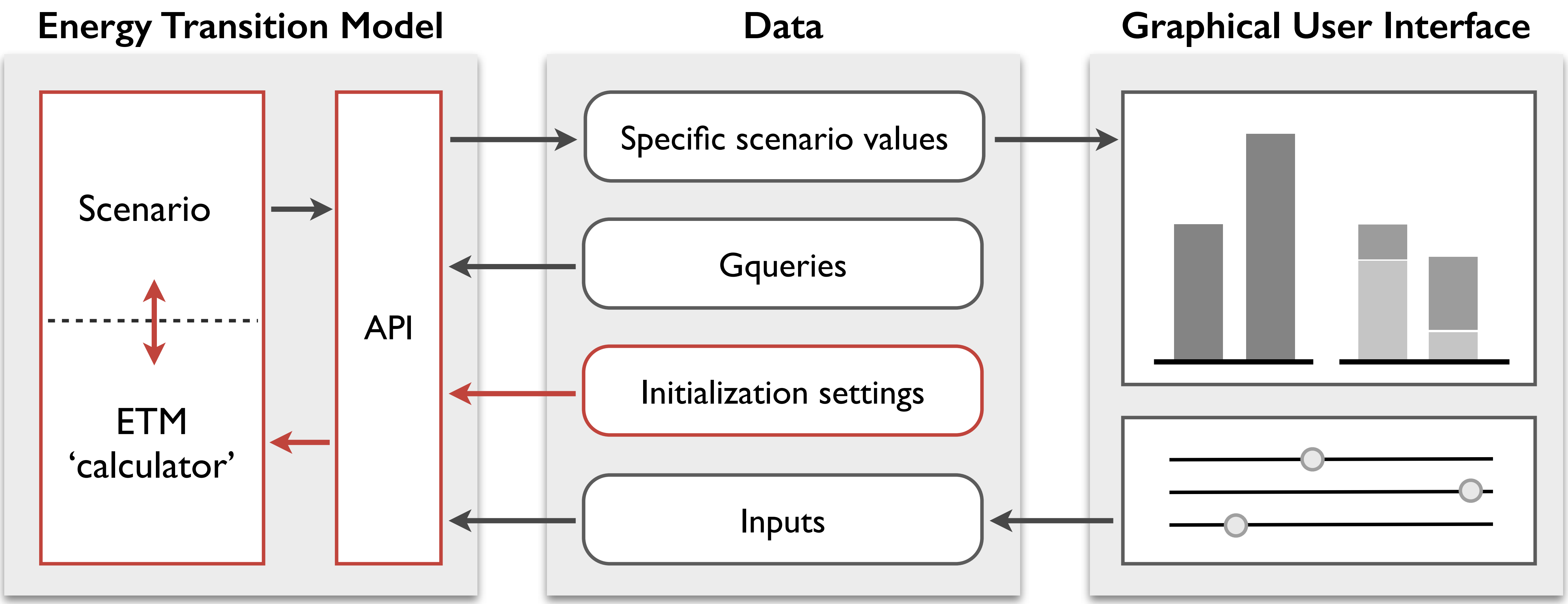


1. **Initialize Sliders**
2. Send Initialization settings
3. Send Gqueries
4. Receive specific scenario values
5. Initialize graphics
6. Moving a slider
7. Update inputs from sliders
8. Send inputs to scenario
9. Send Gqueries to scenario
10. Receive specific scenario values
11. Update graphics

Command `var Sliders1 = new Build_Sliders ({ options });`

Description The object `Build_Sliders` is defined in `ETM_API_example_functions.js` and adds sliders to the slider-holder-div as defined in `ETM_API_example.html`. An instance of the `Build_Sliders` object is created in `ETM_API_example_data.js`.The `options`-object contains information about:
- which holder to append the sliders;
- which scenario the sliders are linked to;
- what function to execute after sliding a slider;
- the `SliderData`.

The `SliderData`-object contains information about what inputs each slider is linked to, what default value, minium, maximum, and step the slider should have, and what name and unit to display. The `SliderData` is also used in step 7.



Energy Transition Model

Scenario

ETM 'calculator'

API

Data

Specific scenario values

Gqueries

Initialization settings

Inputs

Graphical User Interface

1. Initialize Sliders
2. **Send Initialization settings**
3. Send Gqueries
4. Receive specific scenario values
5. Initialize graphics
6. Moving a slider
7. Update inputs from sliders
8. Send inputs to scenario
9. Send Gqueries to scenario
10. Receive specific scenario values
11. Update graphics

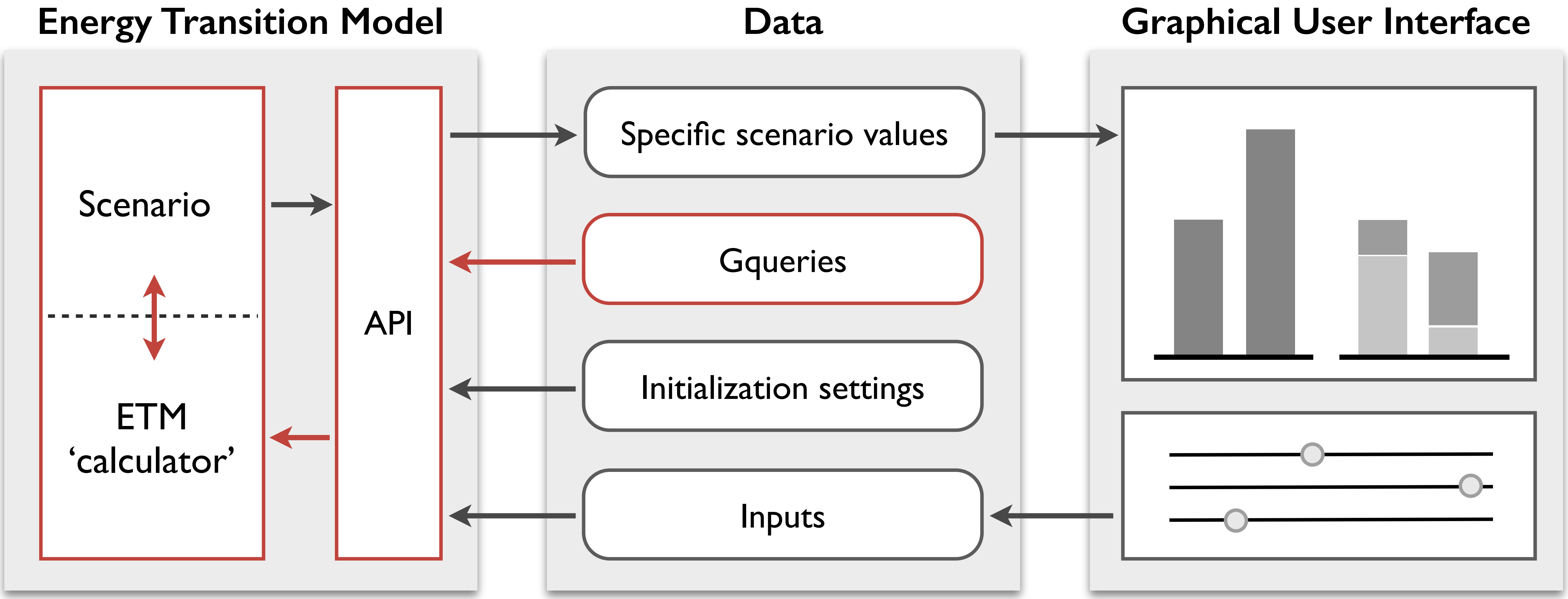
Command

```
var Scenario1 = new ETMscenario ({ options });
```

Description

The object `ETMscenario` is defined in `ETM_API_example_functions.js` and is used for all interactions with the API and the ETM scenario. Upon initializing the user executes the command `ETMscenario.Initialize_Scenario` and sends information about the:
- scenario name, and
- the end year.

The API sends a `responseText` back. This response is used to store the scenario ID in the variable `ETMscenario.Scenario_ID`. This ID is later used to send the inputs and queries to right scenario.

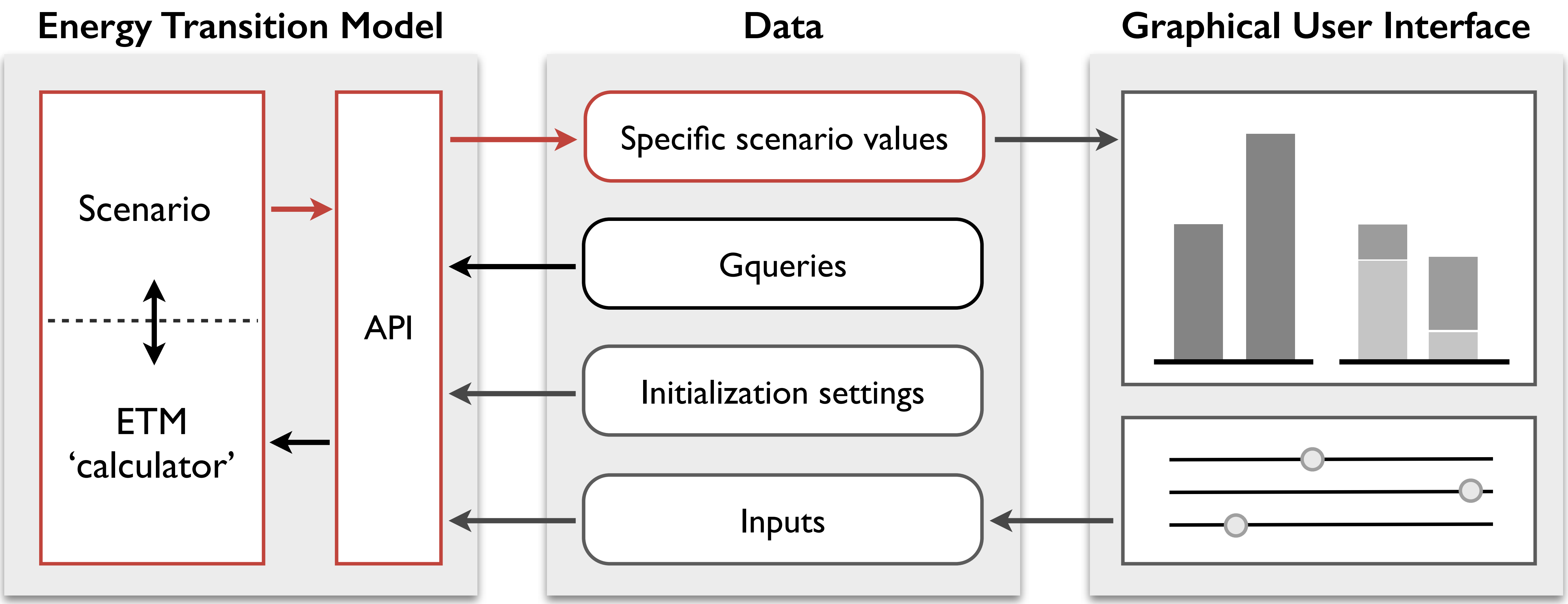


1. Initialize Sliders
2. Send Initialization settings
3. **Send Gqueries**
4. Receive specific scenario values
5. Initialize graphics
6. Moving a slider
7. Update inputs from sliders
8. Send inputs to scenario
9. Send Gqueries to scenario
10. Receive specific scenario values
11. Update graphics

Command `Scenario1.Update_Gqueries();`

Description `Update_Gqueries` is a method of `ETMscenario`. It sends the Gqueries, as listed in `ETM_API_example_data.js`, to the API.

The `ETMscenario.Update_Gqueries` method is executed from within the `ETMscenario.Update_Scenario` method. This method first imports sends the current slider values to the API (steps 7 & 8), but during the initialization these steps don't have an effect as the sliders have the default values.



Energy Transition Model

Scenario

ETM
‘calculator’

API

Data

Specific scenario values

Gqueries

Initialization settings

Inputs

Graphical User Interface

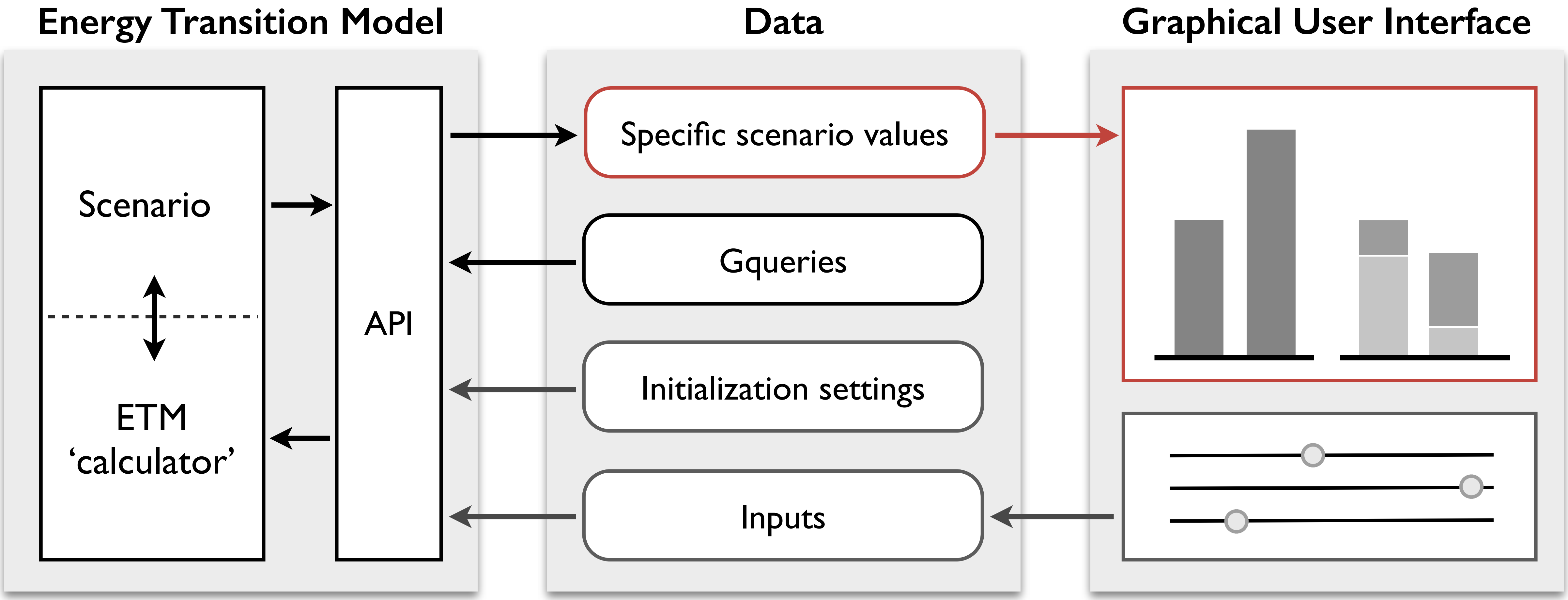
1. Initialize Sliders
2. Send Initialization settings
3. Send Gqueries
4. **Receive specific scenario values**
5. Initialize graphics
6. Moving a slider
7. Update inputs from sliders
8. Send inputs to scenario
9. Send Gqueries to scenario
10. Receive specific scenario values
11. Update graphics

Command `Scenario1.Update_Gqueries();`

Description The method `ETMscenario.Update_Gqueries` stores the specific scenario values in the object `ETMscenario.Data`.This Data object is structured as:

```
ETMscenario.Data = {
 NAME1: {"present": [present value],
         "future": [future value]},
 NAME2: {"present": [present value],
         "future": [future value]},
  ...
}
```

The names are given in the `ETM_API_example_data.js` file. In this example, the data object is used to make charts, the values could be used for many other purposed (textual, graphics, etc.)



1. Initialize Sliders
2. Send Initialization settings
3. Send Gqueries
4. Receive specific scenario values
5. **Initialize graphics**
6. Moving a slider
7. Update inputs from sliders
8. Send inputs to scenario
9. Send Gqueries to scenario
10. Receive specific scenario values
11. Update graphics

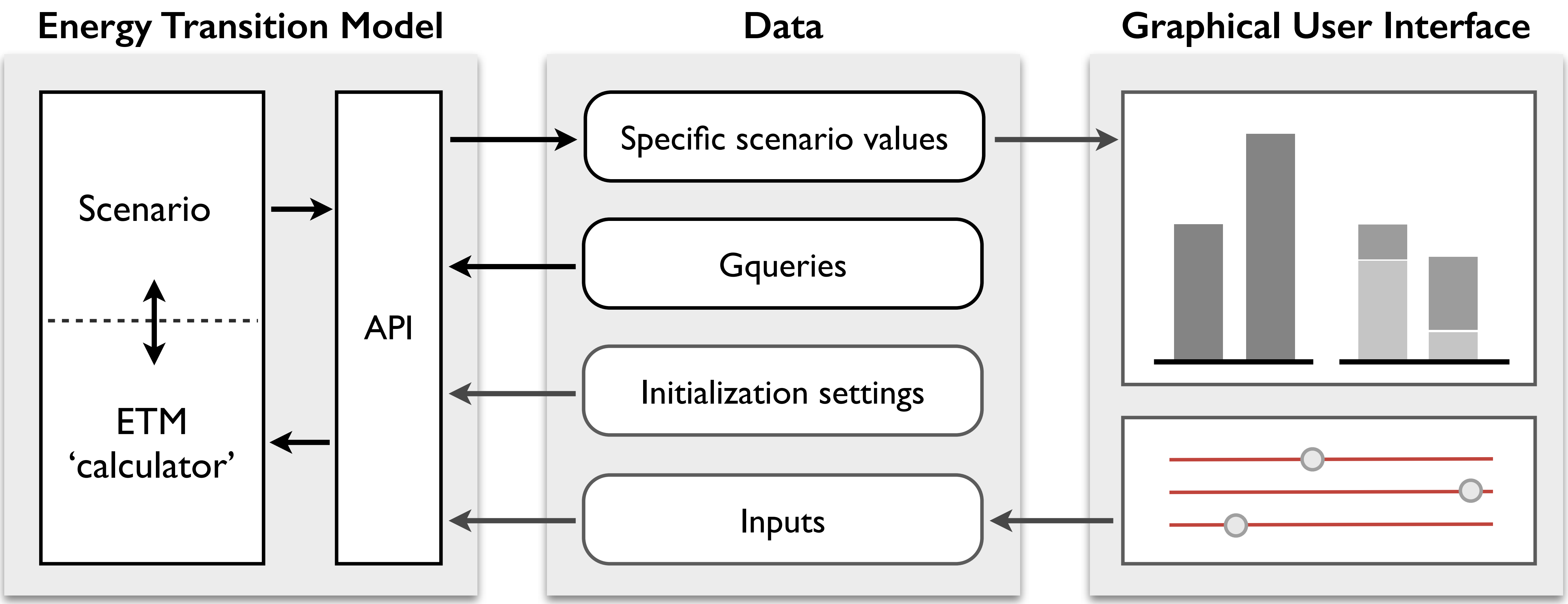
Command

```
$( '#chart_ID' ).highcharts({ options })
```

Description

In this example the charts are build with the highcharts.js library. In the file `ETM_API_example_graphs.js` all the code is shown to build the three different graphs in the example. The three graphs are individually initialized by the `.highcharts({options})` function.

An important step here is building the data series. There is no default way to put Gquery values in a chart, so here the user really has to decided what values to group together and how to display them. For the example charts, the data series are built with the functions `HeatSeries`, `PrimarySeries` and `CO2series`.



Energy Transition Model

Scenario

ETM 'calculator'

API

Data

Specific scenario values

Gqueries

Initialization settings

Inputs

Graphical User Interface

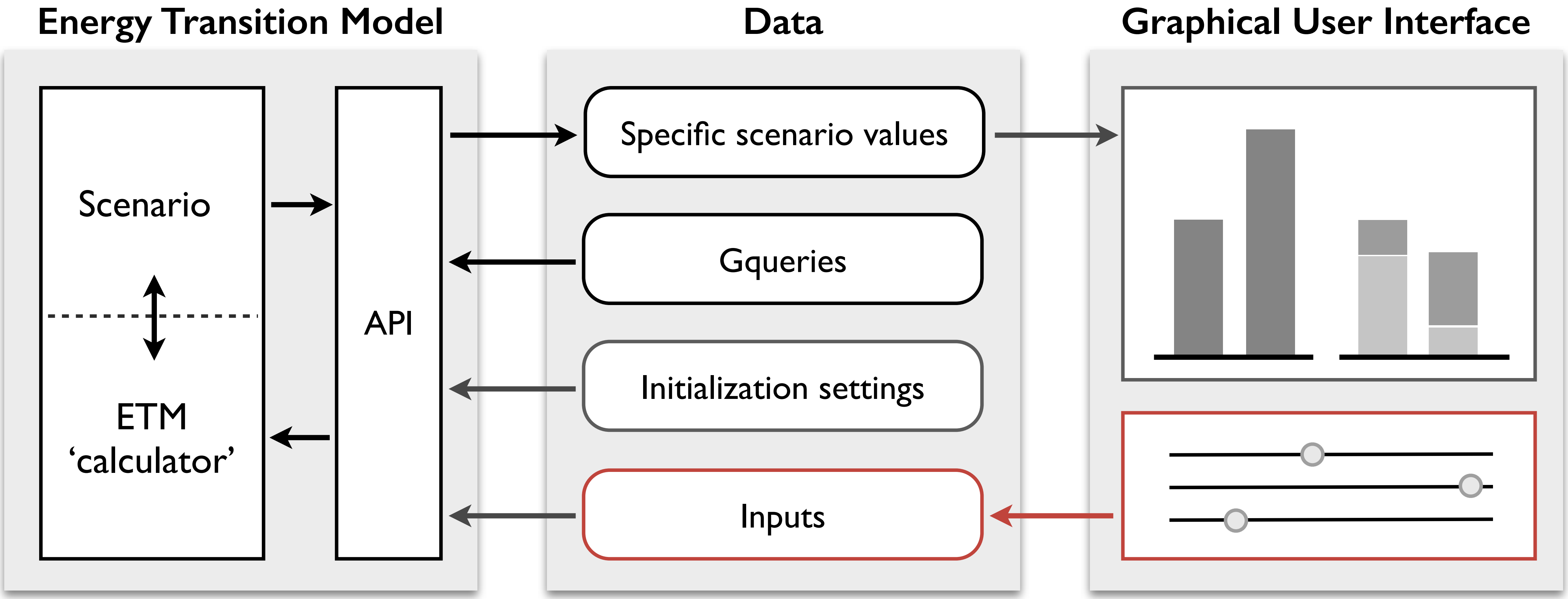
1. Initialize Sliders
2. Send Initialization settings
3. Send Gqueries
4. Receive specific scenario values
5. Initialize graphics
6. **Moving a slider**
7. Update inputs from sliders
8. Send inputs to scenario
9. Send Gqueries to scenario
10. Receive specific scenario values
11. Update graphics

Command `on stop: eval(options.UpdateFunction)`

Description During the initialization of the sliders (step 1), the behaviour of the slider *during* the slide and on a slide *stop* is defined (within the method `Build_Sliders.SliderInitialize`).

During a slide the slider value display is updated so that the user knows what value the slider is on.

After a slide-stop, the method `SliderInitialize.options.UpdateFunction` is executed. This function, in turn, executes the method `Scenario1.Update_Scenario()` and the function `Update_Graphs`, leading to steps 7-11.



Energy Transition Model

Scenario

ETM 'calculator'

API

Data

Specific scenario values

Gqueries

Initialization settings

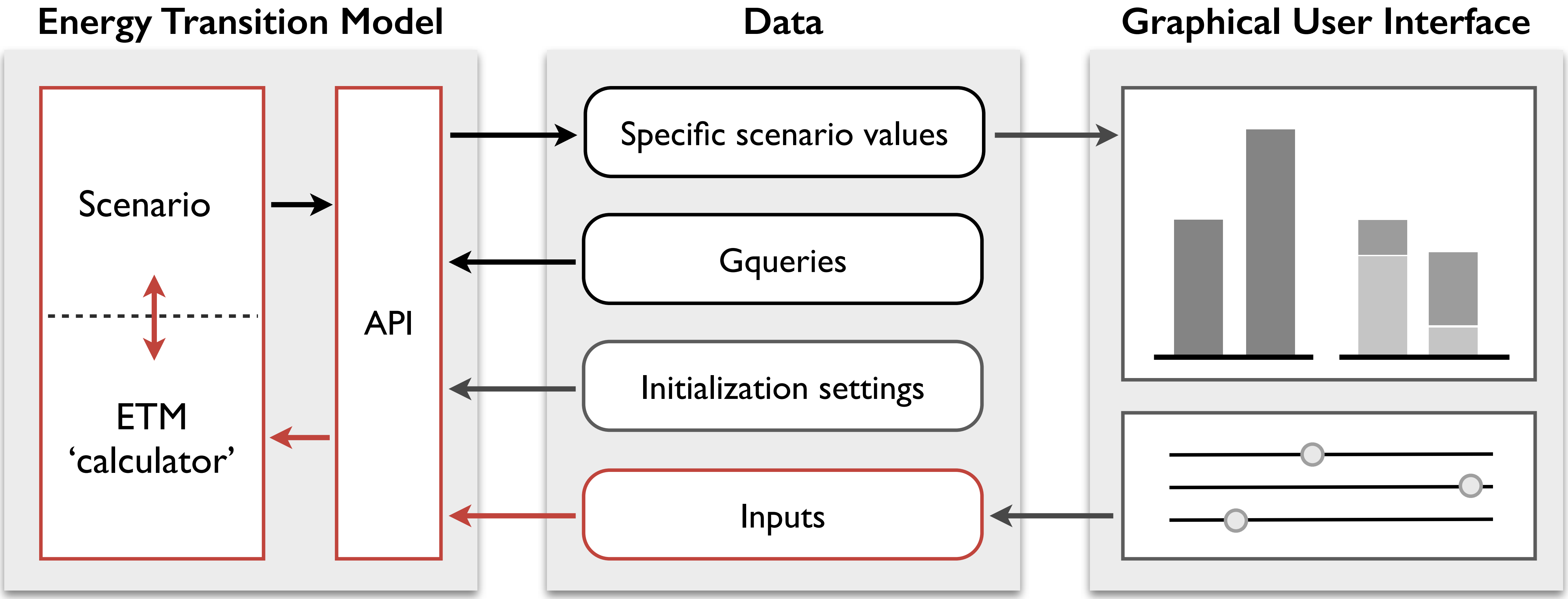
Inputs

Graphical User Interface

1. Initialize Sliders
2. Send Initialization settings
3. Send Gqueries
4. Receive specific scenario values
5. Initialize graphics
6. Moving a slider
7. **Update inputs from sliders**
8. Send inputs to scenario
9. Send Gqueries to scenario
10. Receive specific scenario values
11. Update graphics

Command `ETMscenario.Update_Inputs_From_Sliders();`

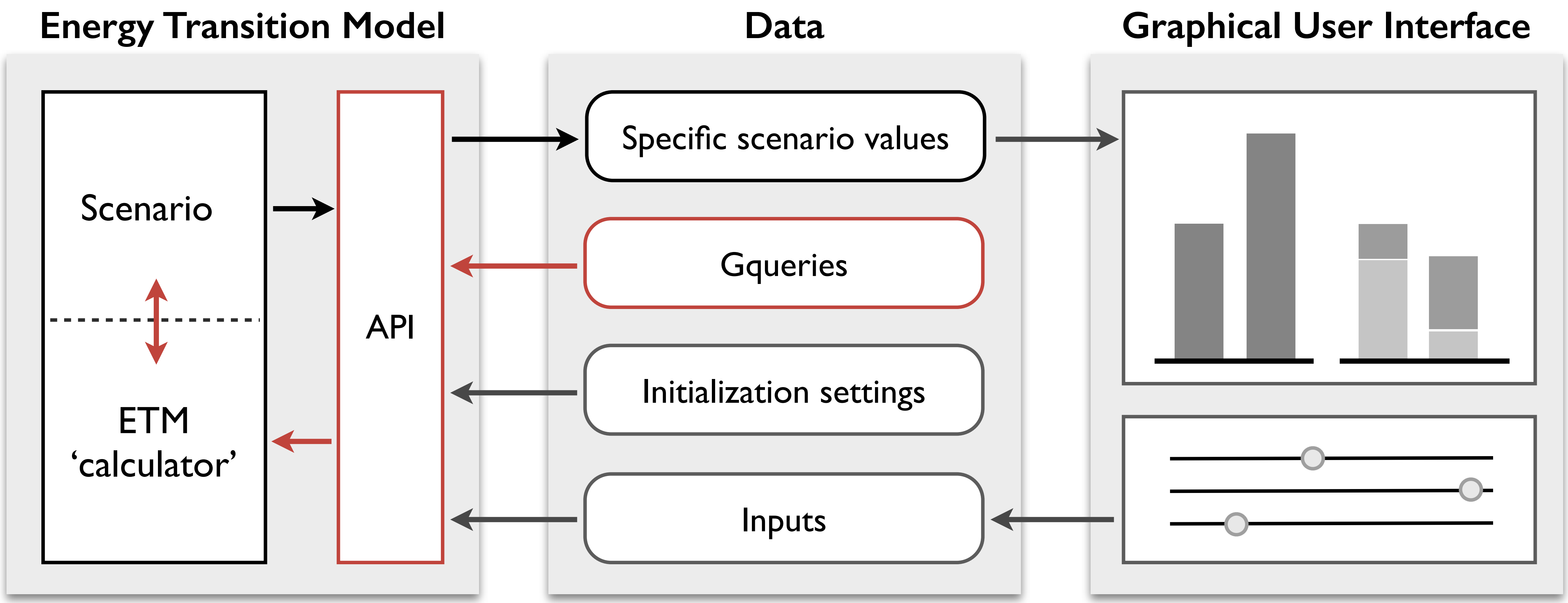
Description The method `ETMscenario.Update_Inputs_From_Sliders()` uses jQuery to fill the object `ETMscenario.Input_Settings` with the slidervalues. To make this work, the `Inputs` object that is passed within the options object while conscrucing `Scenario1` (step 2) needs to contain the same names as the object keys in `SliderData`, part of the options object while constructing `Sliders1` (step 1).



1. Initialize Sliders
2. Send Initialization settings
3. Send Gqueries
4. Receive specific scenario values
5. Initialize graphics
6. Moving a slider
7. Update inputs from sliders
8. **Send inputs to scenario**
9. Send Gqueries to scenario
10. Receive specific scenario values
11. Update graphics

Command `ETMscenario.Update_Inputs();`

Description The method `ETMscenario.Update_Inputs()` sends the input settings from step 7 to the API. The `Scenario_ID` is used to send the inputs to the right scenario.



Energy Transition Model

Scenario

ETM 'calculator'

API

Data

Specific scenario values

Gqueries

Initialization settings

Inputs

Graphical User Interface

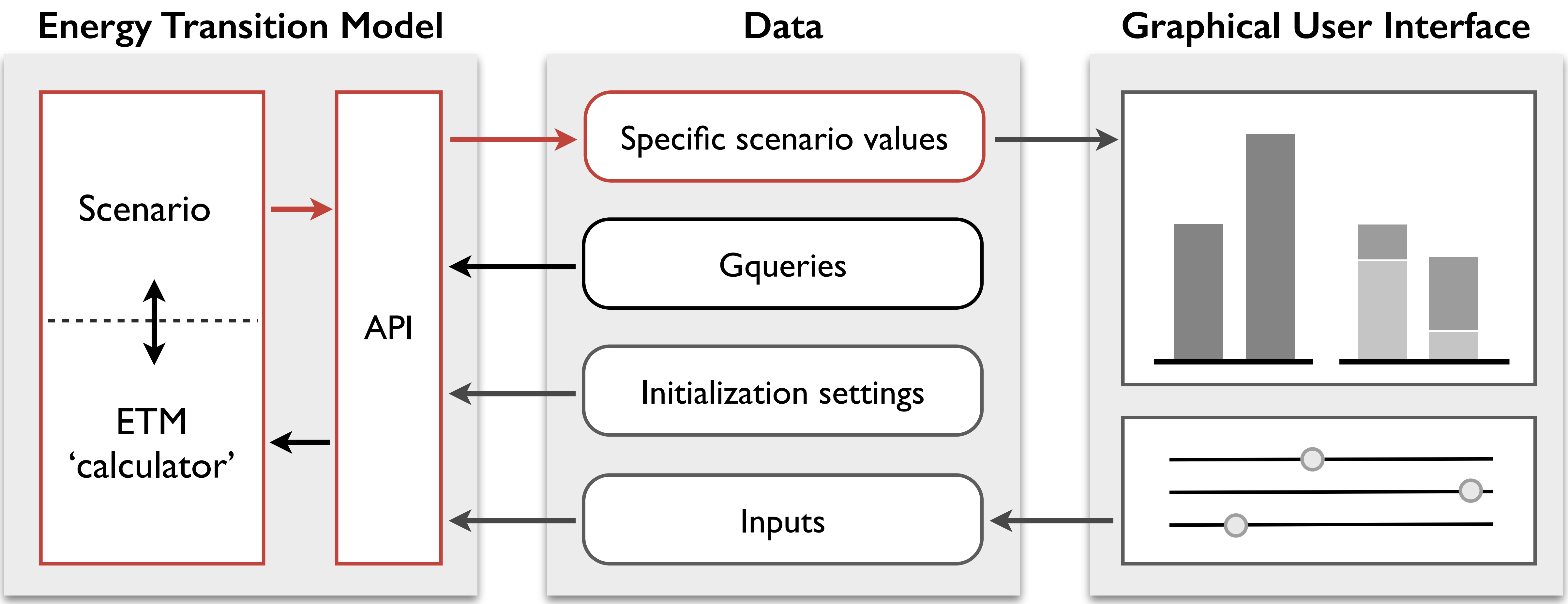
1. Initialize Sliders
2. Send Initialization settings
3. Send Gqueries
4. Receive specific scenario values
5. Initialize graphics
6. Moving a slider
7. Update inputs from sliders
8. Send inputs to scenario
9. **Send Gqueries to scenario**
10. Receive specific scenario values
11. Update graphics

Command `ETMscenario.Update_Gqueries();`

Description This step is identical to step 3.

`Update_Gqueries` is a method of `ETMscenario`. It sends the Gqueries, as listed in `ETM_API_example_data.js`, to the API.

The `ETMscenario.Update_Gqueries` method is executed from within the `ETMscenario.Update_Scenario` method. This method first imports sends the current slider values to the API (steps 7 & 8), but during the initialization these steps don't have an effect as the sliders have the default values.



Energy Transition Model

Scenario

ETM 'calculator'

API

Data

Specific scenario values

Gqueries

Initialization settings

Inputs

Graphical User Interface

1. Initialize Sliders
2. Send Initialization settings
3. Send Gqueries
4. Receive specific scenario values
5. Initialize graphics
6. Moving a slider
7. Update inputs from sliders
8. Send inputs to scenario
9. Send Gqueries to scenario
10. **Receive specific scenario values**
11. Update graphics

Command `ETMscenario.Update_Gqueries();`

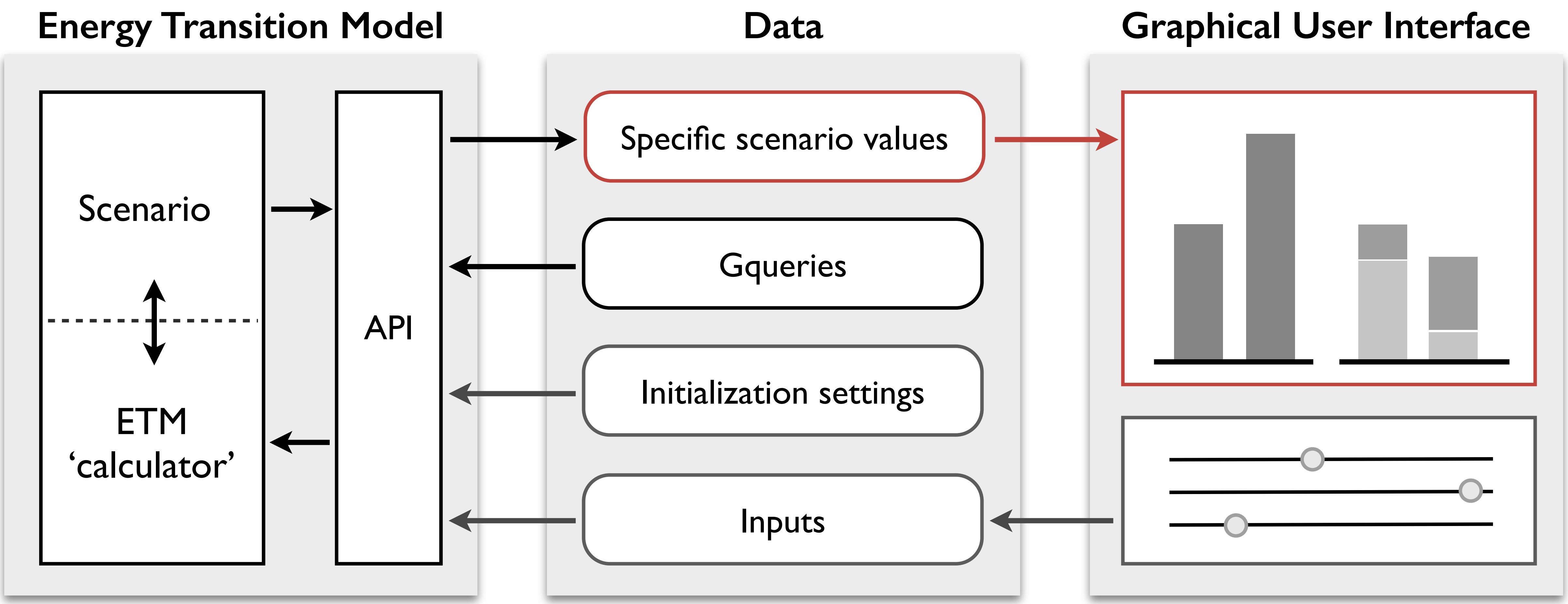
Description

This step is identical to step 4.

The method `ETMscenario.Update_Gqueries` stores the specific scenario values in the object `ETMscenario.Data`.This Data object is structured as:

```
ETMscenario.Data = {
 NAME1: {"present": [present value],
         "future": [future value]},
 NAME2: {"present": [present value],
         "future": [future value]},
  ...
}
```

The names are given in the `ETM_API_example_data.js` file. In this example, the data object is used to make charts, the values could be used for many other purposed (textual, graphics, etc.)



Energy Transition Model

Scenario

ETM ‘calculator’

API

Data

Specific scenario values

Gqueries

Initialization settings

Inputs

Graphical User Interface

1. Initialize Sliders
2. Send Initialization settings
3. Send Gqueries
4. Receive specific scenario values
5. Initialize graphics
6. Moving a slider
7. Update inputs from sliders
8. Send inputs to scenario
9. Send Gqueries to scenario
10. Receive specific scenario values
11. **Update graphics**

Command

```
Update_Graphs( 250 );
```

Description

The function `Update_Graphs` is defined in `ETM_API_example_graphs.js` and uses the `ETMscenario.Data` object to update the charts. To do this, it uses the built-in highcharts method `setData`, with a transition time `transtime`. The `setData` method is executed from within the function `seriesupdate`, that updates all data series withing a specific chart.