MATLAB EXPO 2021

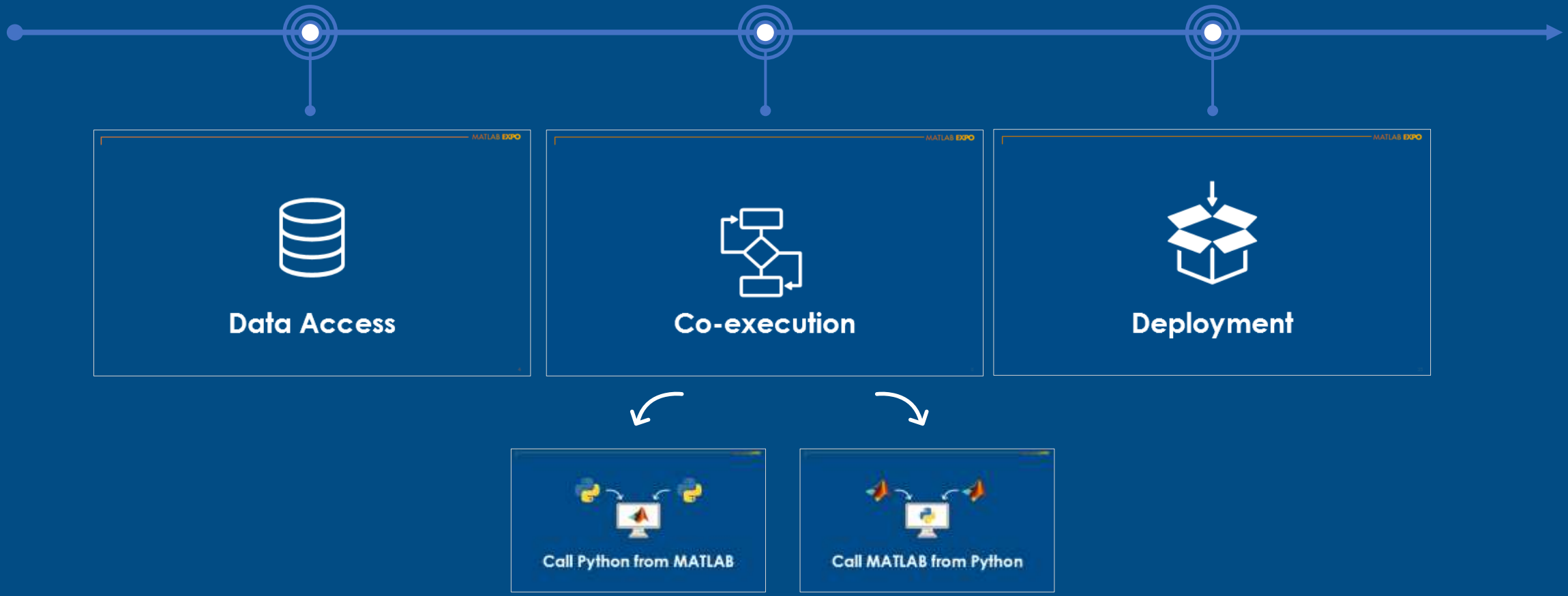
MATLAB With Python *for engineers and data scientists*

성 호 현 The MathWorks Korea

MathWorks

Example: Build Air Quality App using MATLAB and Python

Development plan



Data Access

Access Data from a Web Service

https://openweathermap.org/

Data Access

Co-Execution

- Call Python from MATLAB
- Call MATLAB from Python

Deployment

OpenWeather global services

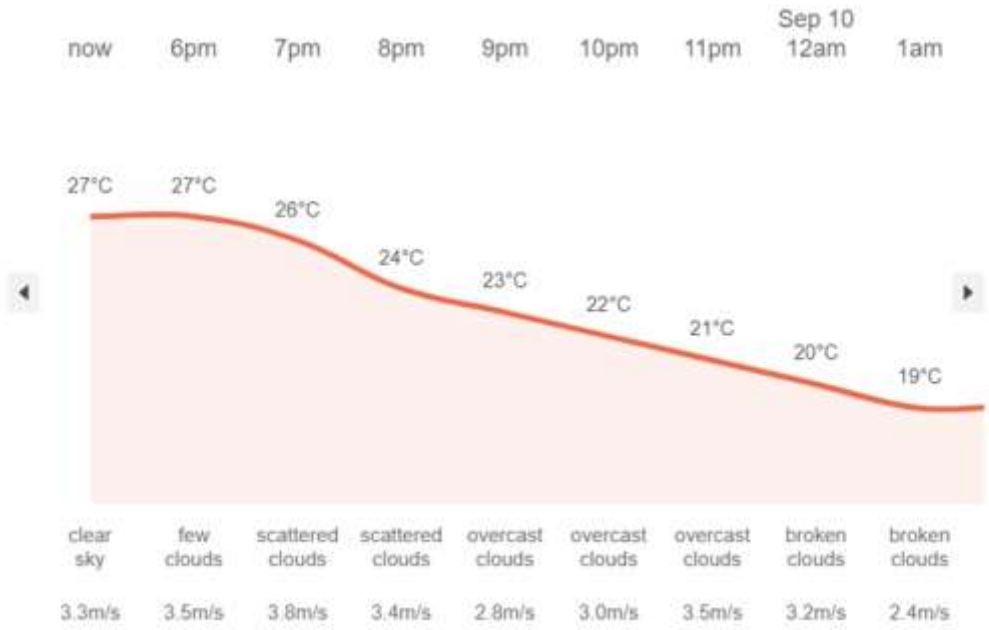
Weather forecasts, nowcasts and history in fast and elegant way

2 Billion Forecasts Per Day
2,500 new subscribers a day

2,600,000 customers
20+ weather APIs

Hourly forecast

Temperature | Precipitation



8-day forecast

Wed, Sep 09	27 / 19°C	clear sky
Thu, Sep 10	25 / 18°C	broken clouds
Fri, Sep 11	28 / 17°C	scattered clouds
Sat, Sep 12	25 / 16°C	clear sky
Sun, Sep 13	26 / 15°C	broken clouds
Mon, Sep 14	32 / 17°C	scattered clouds
Tue, Sep 15	27 / 20°C	moderate rain
Wed, Sep 16	21 / 17°C	light rain

What type of data?

Numerical, Textual, Geolocalized, Timeseries, …

Called by:
geographical coordinates, zip codes, city name, city ID, number of cities (only in current and forecasted APIs)

https://openweathermap.org/

Store & transfer tabular data between languages

Parquet files

- Columnar storage format available to any project in the Hadoop big data ecosystem, regardless of the choice of data processing framework, data model or programming language (More on Parquet)

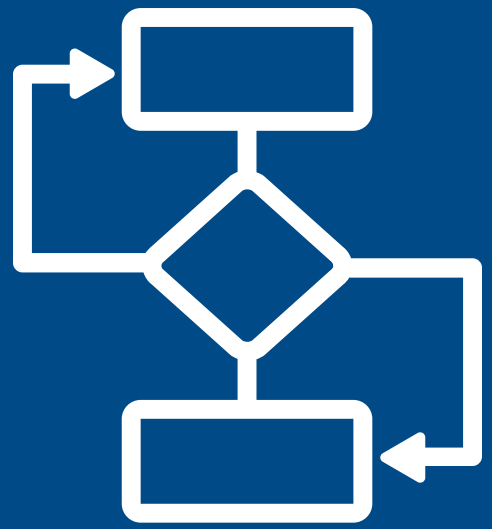

```
parquetwrite("temperatureFitting.parquet",T)
```

T = 40×3 table

	Time	Temperatures	SineFit
1	2020-07-16…	19.0200	23.8001
2	2020-07-16…	20.3300	24.6581
3	2020-07-16…	20.1300	23.0750
4	2020-07-16…	18.9300	19.9678
5	2020-07-17…	17.1500	17.1366
6	2020-07-17…	16.0200	16.2214
7	2020-07-17…	16.7900	17.7525

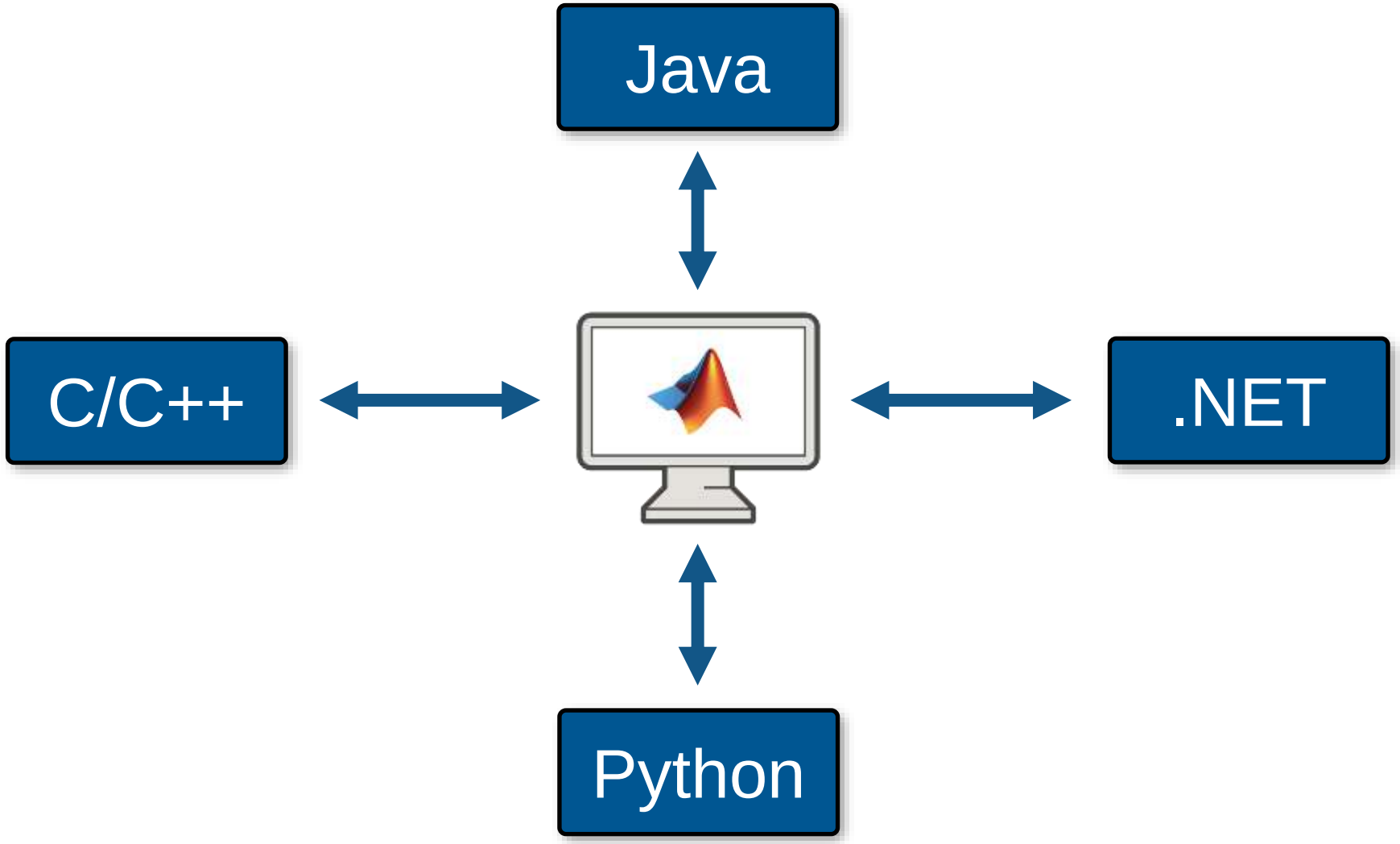
```
# Use parquet file as alternative to exchange tables with MATLAB
df = pd.read_parquet("temperatureFitting.parquet")
df.head()
```

	Time	Temperatures	SineFit
0	2020-07-16 12:00:00	19.02	23.800150
1	2020-07-16 15:00:00	20.33	24.658138
2	2020-07-16 18:00:00	20.13	23.074984
3	2020-07-16 21:00:00	18.93	19.967804
4	2020-07-17 00:00:00	17.15	17.136577



Co-execution

MATLAB provides flexible integration with multiple languages



Given: Existing Python Code accessing & preparing weather data

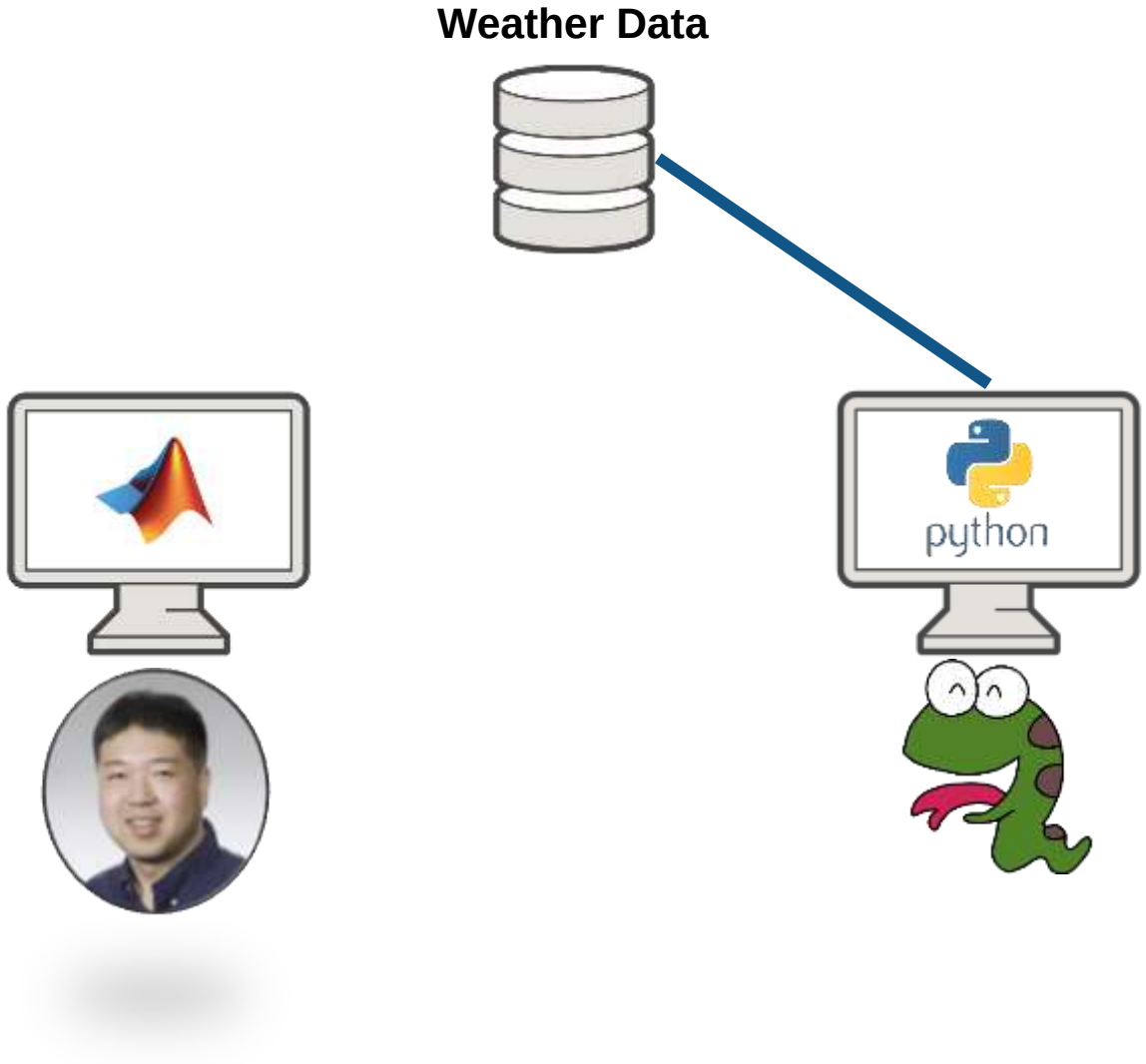
Data preparation | Modeling | Deployment

Data Access

Co-Execution

- Call Python from MATLAB
- Call MATLAB from Python

Deployment



Call Python from MATLAB

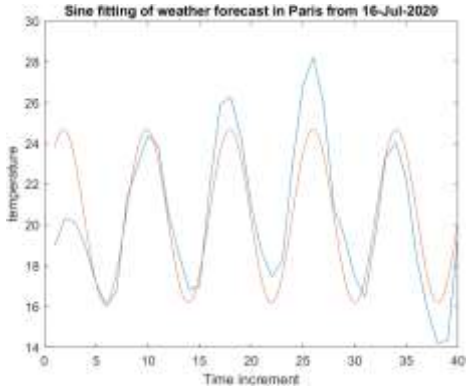
Data preparation | Modeling | Deployment

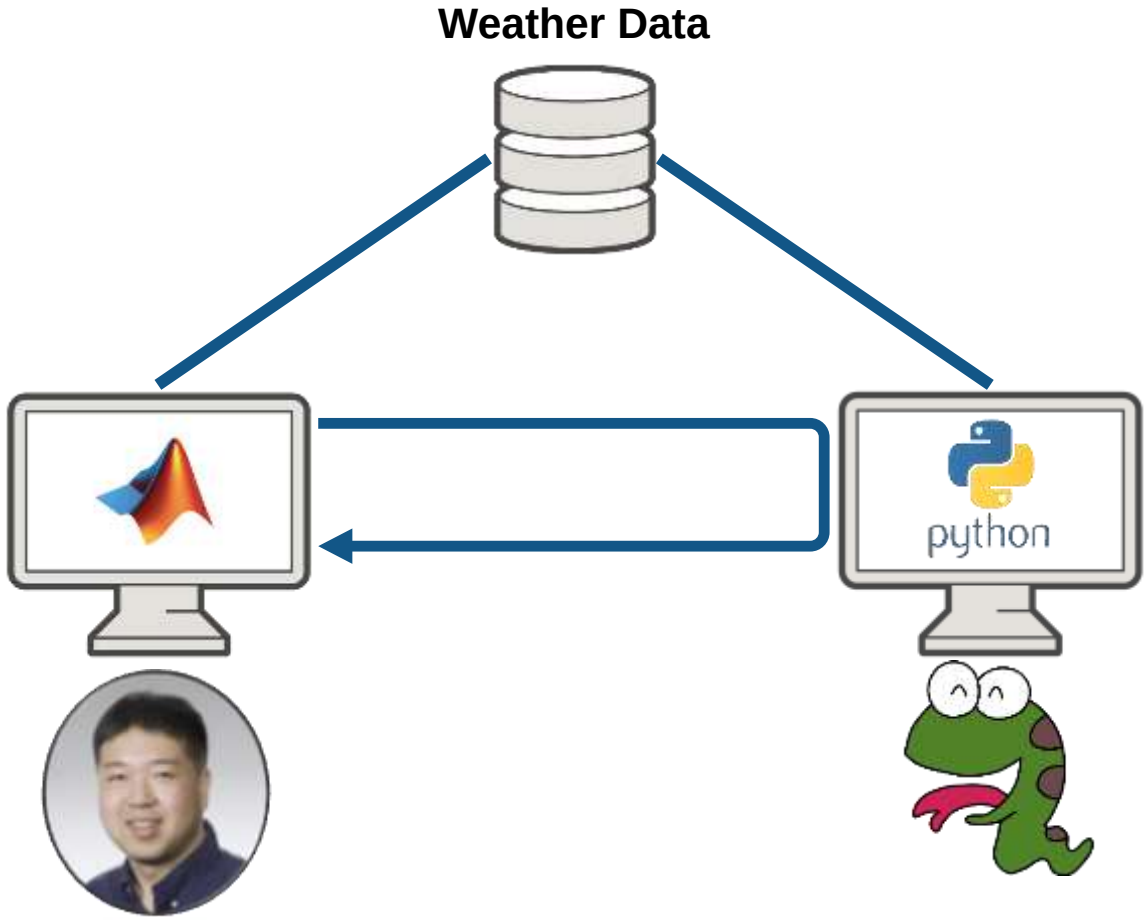
Data Access

Co-Execution

- Call Python from MATLAB
- Call MATLAB from Python

Deployment





Call MATLAB from Python

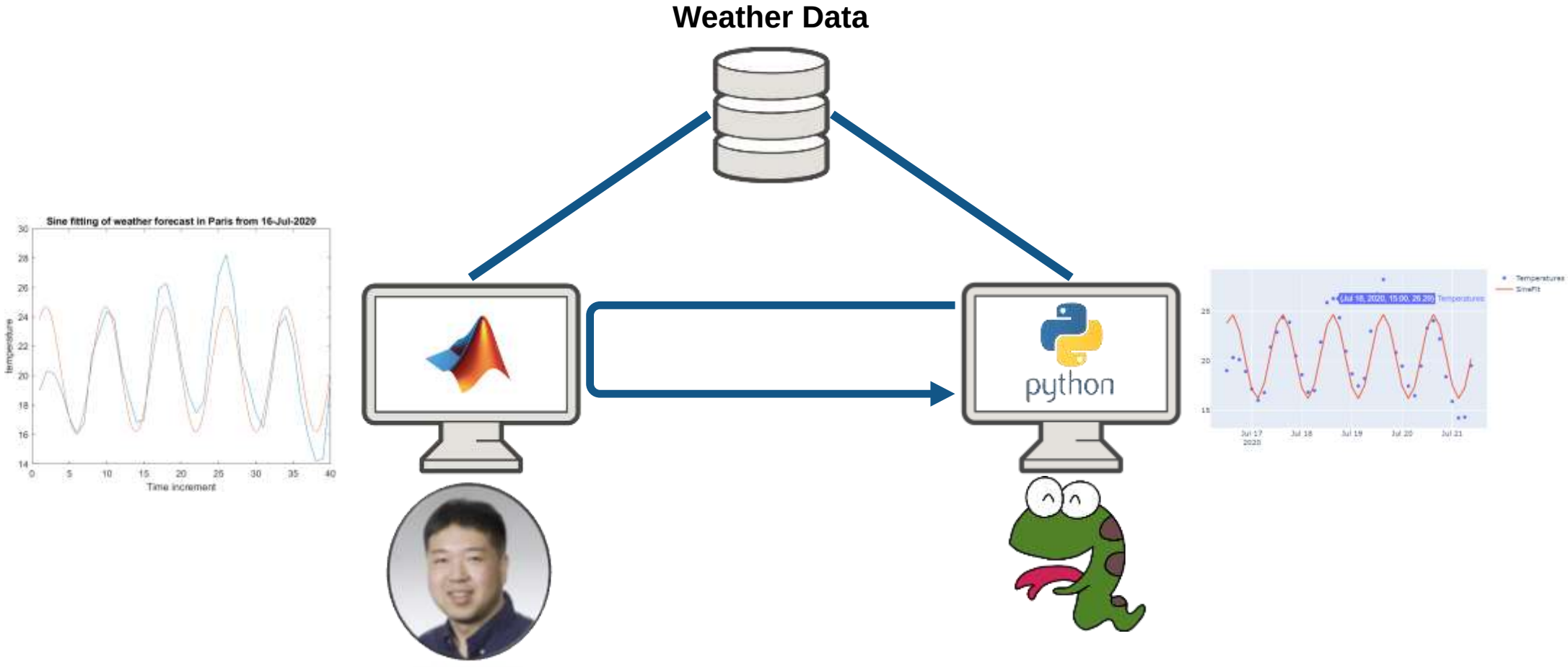
Data preparation | Modeling | Deployment

Data Access

Co-Execution

- Call Python from MATLAB
- Call MATLAB from Python

Deployment



Deploy:
MATLAB Analytics into Python

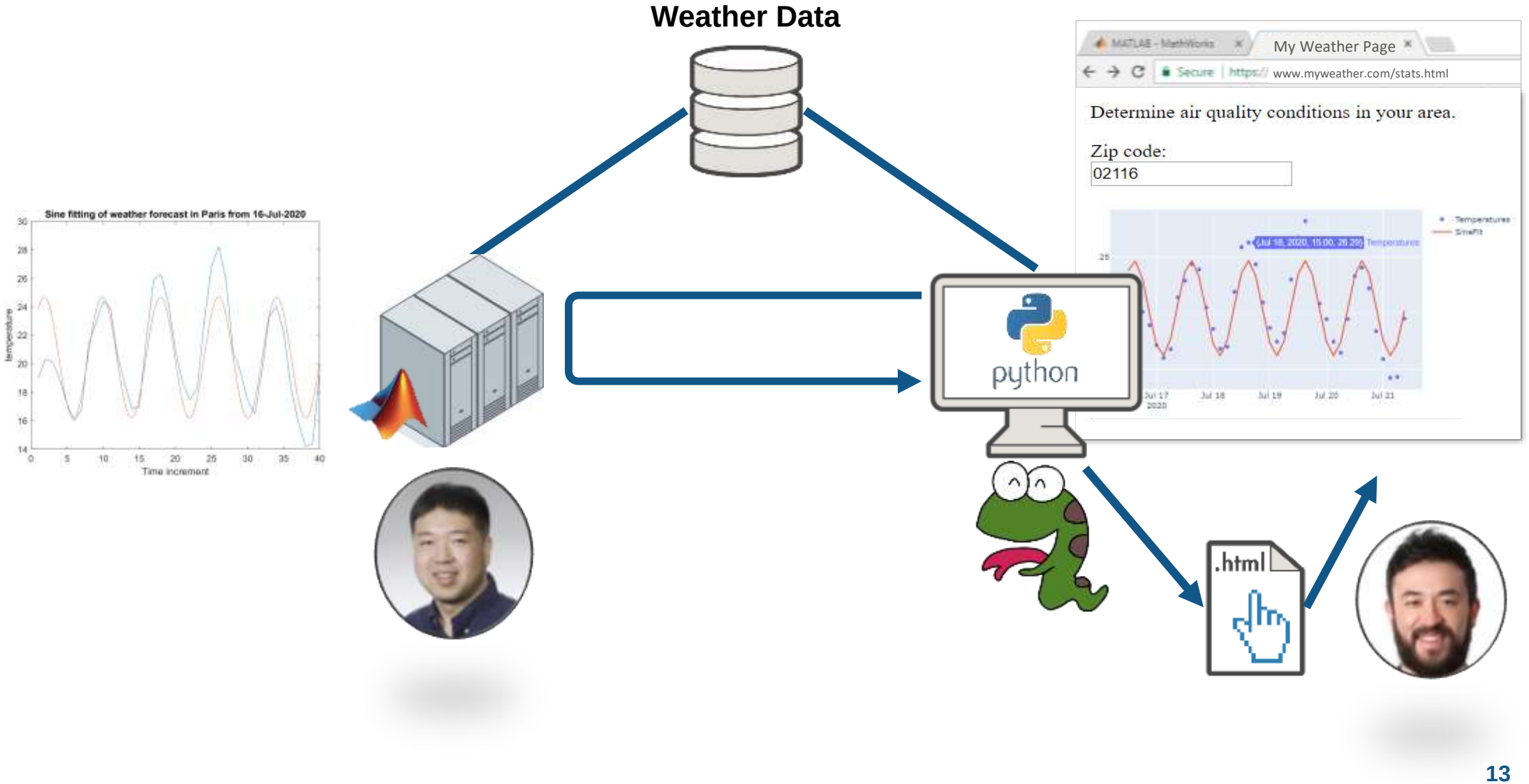
Data preparation → Modeling → Deployment

Data Access

Co-Execution

- Call Python from MATLAB
- Call MATLAB from Python

Deployment



Deploy:
MATLAB Analytics into Python

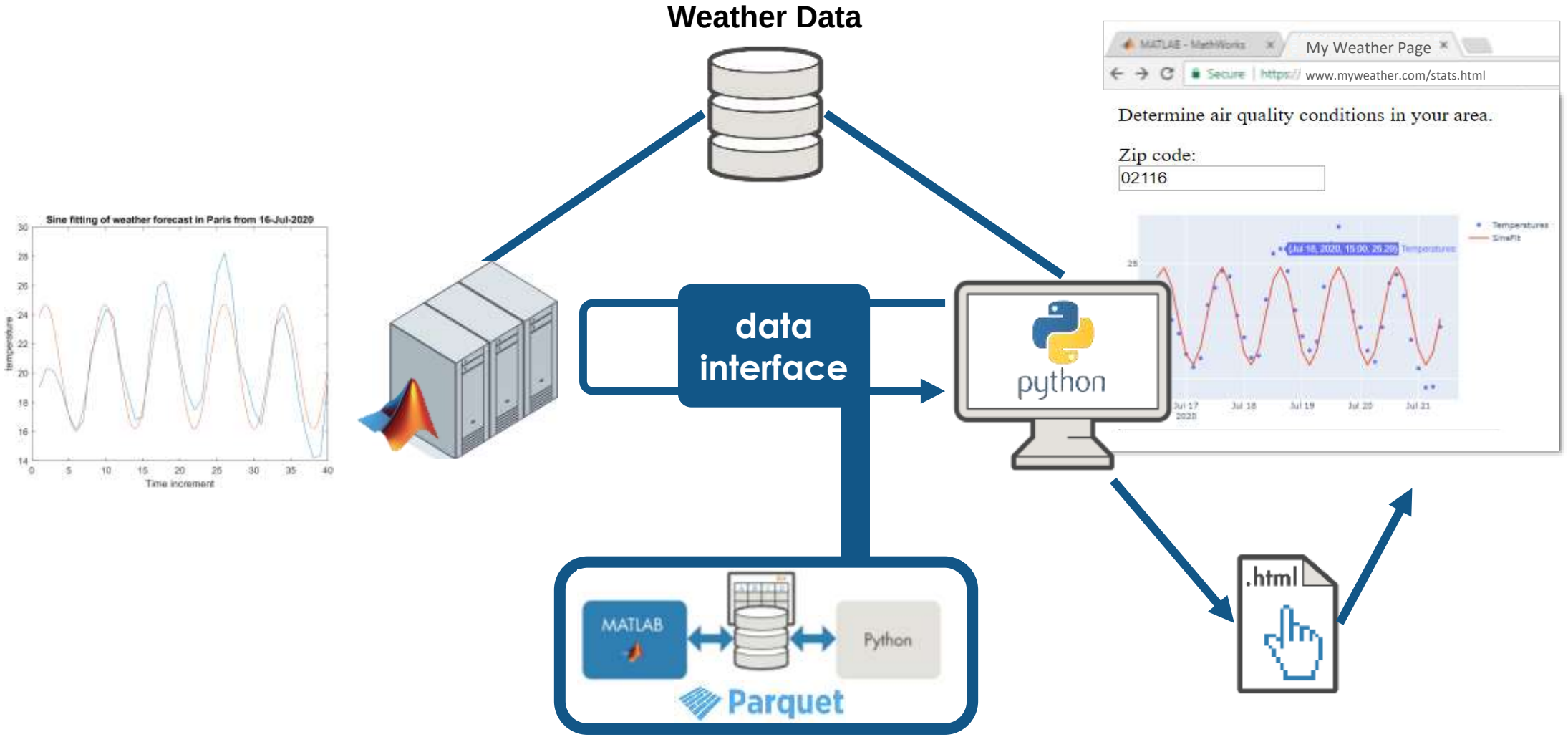
Data preparation → Modeling → Deployment

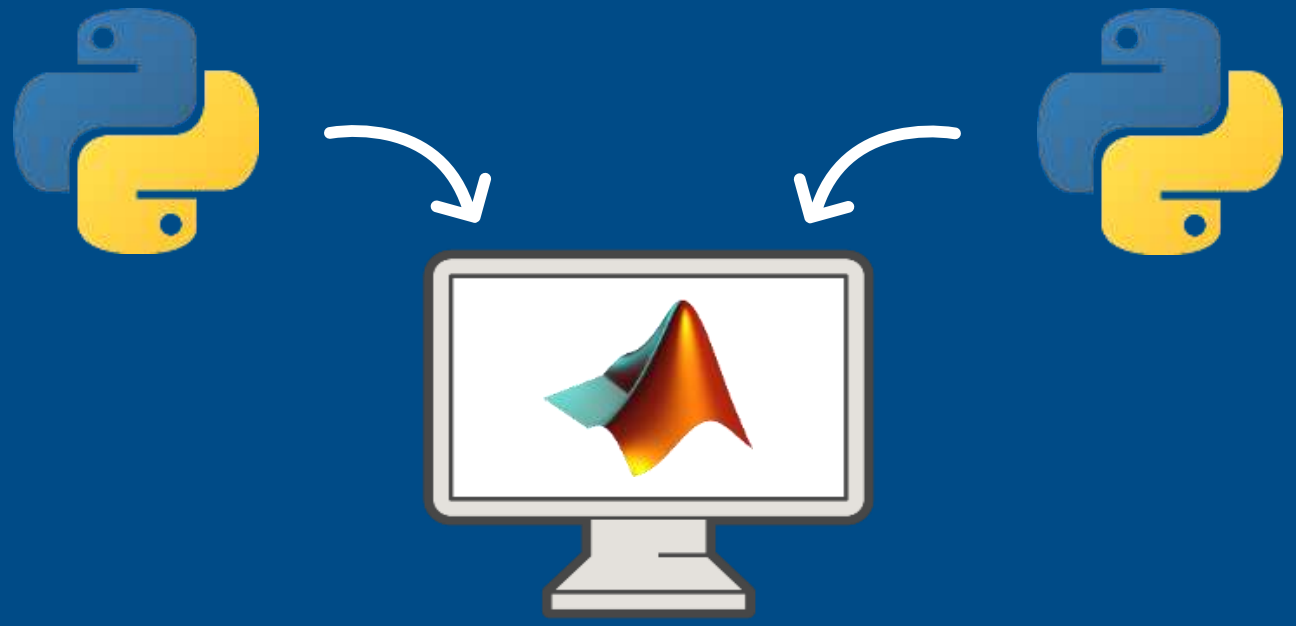
Data Access

Co-Execution

- Call Python from MATLAB
- Call MATLAB from Python

Deployment



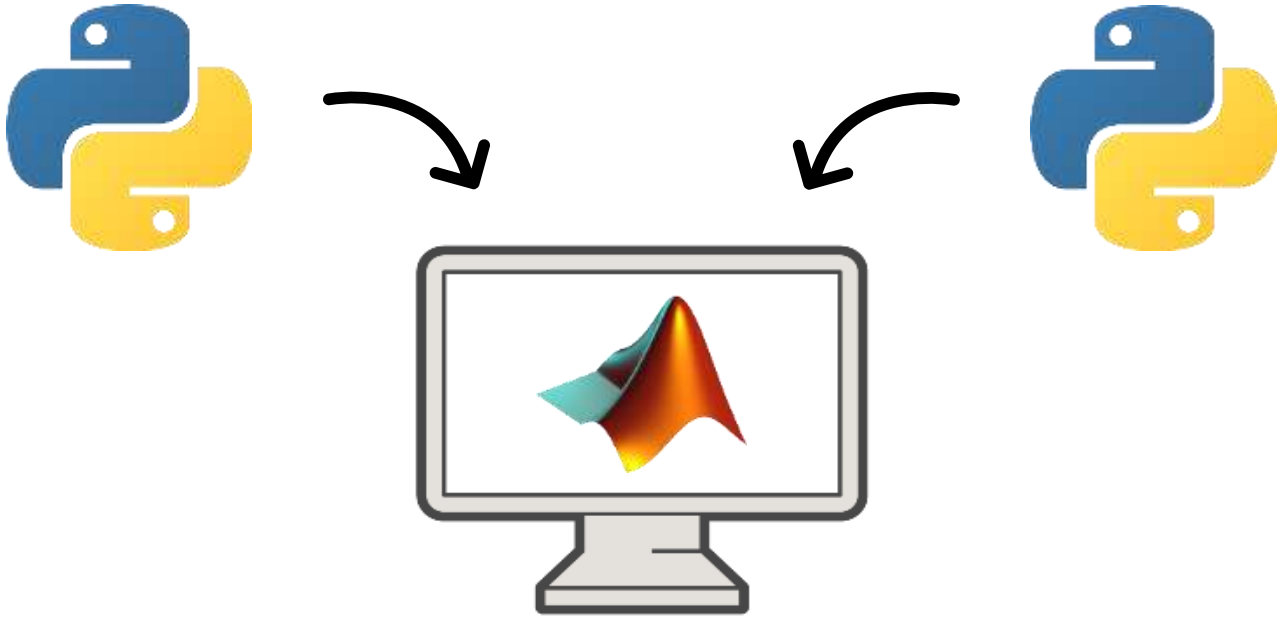


Call Python from MATLAB

Why Call Python from MATLAB?

Already working in MATLAB, and:

- Want to reuse existing Python code
- Need functionality that is only available in Python



Calling Python libraries from MATLAB

Use the weather.py module to get the air quality for Paris. This is a user-defined Python module which includes functions to read and parse the current and forecasted weather data by location.

```
jsonData = py.weather.get_current_weather("Paris","France",apikey.Key)
```

```
jsonData =
  Python dict with no properties.

    {'coord': {'lon': 2.35, 'lat': 48.85}, 'weather': [{'id': 803, 'main': 'Cloud
```

```
def get_current_weather(city, country, apikey):
    # get current conditions in specified location
    # get_current_weather('boston','us',key)
    import urllib.request
    import json
    # read current conditions
    try:
        url = "https://api.openweathermap.org/data/2.5/weather?q="+city+","+country+"&appid="+apikey
        response = urllib.request.urlopen(url)
        html = response.read()
        json_data = json.loads(html)

    except urllib.error.URLError:
        # if weather API doesnt work, read the file
        json_data = read_backup(city)

    return json_data
```

Parse the json data returned from the weather API.

The Python dictionary can be represented as a MATLAB struct.

```
weatherData = py.weather.parse_json(jsondata);
struct(weatherData)
```

```
ans = struct with fields:
            temp: 18.7100
      feels_like: 17.3000
        temp_min: 17.7800
        temp_max: [1×1 py.int]
```

Use a function (`prepData.m`) to prepare data for machine learning (create a table with the expected variable names, preprocessing steps, etc).

```
currentData = prepData(weatherData)
```

currentData = 1×12 table

	DateLocal	city	StateName	T	P	DP	RH	WindDir	WindSpd
1	01-Jul-2020 11:...	"Paris"	Ile de France	21.6200	20.2600	349.2200	1010	5.1000	73

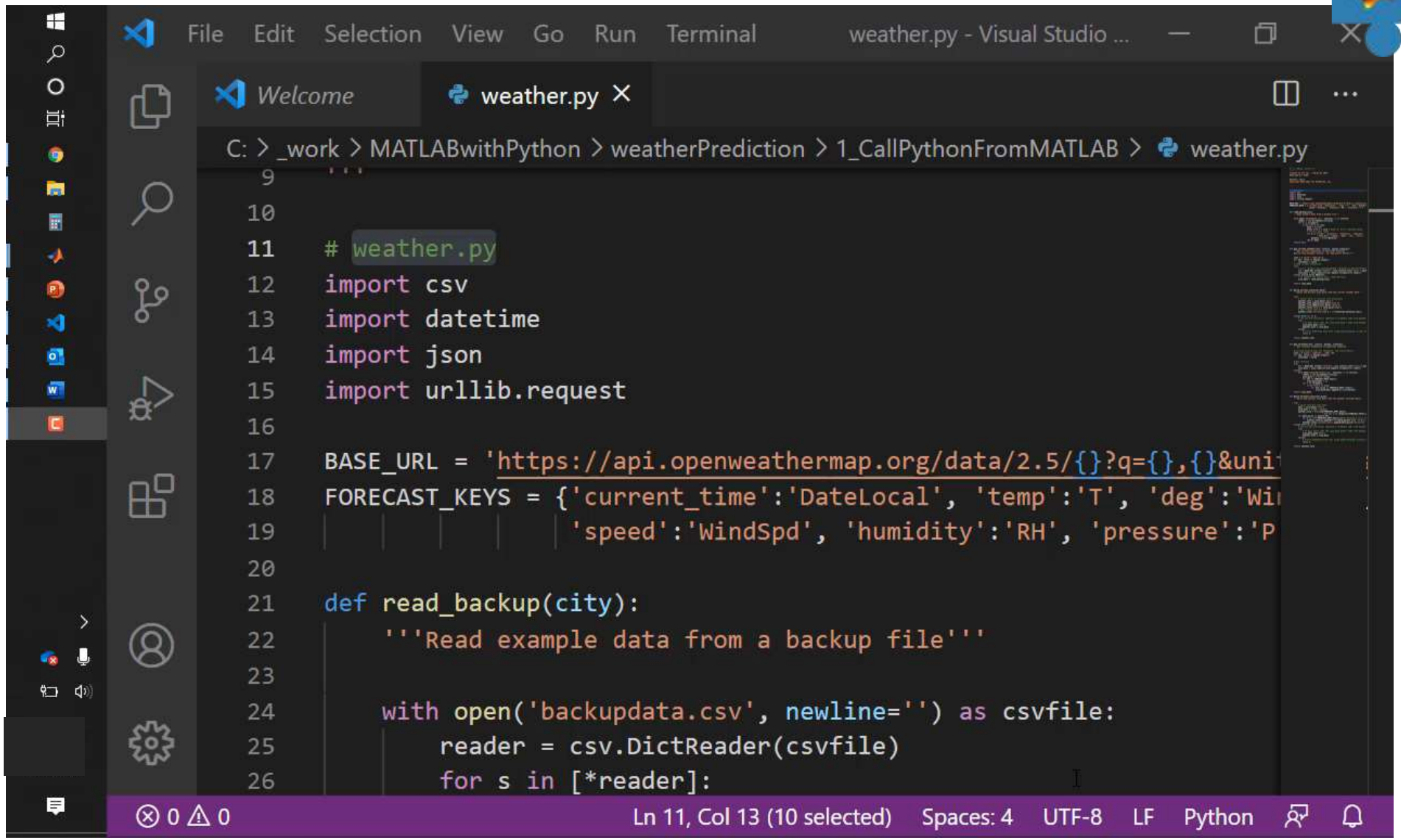
Calling Python libraries from MATLAB

Data Access

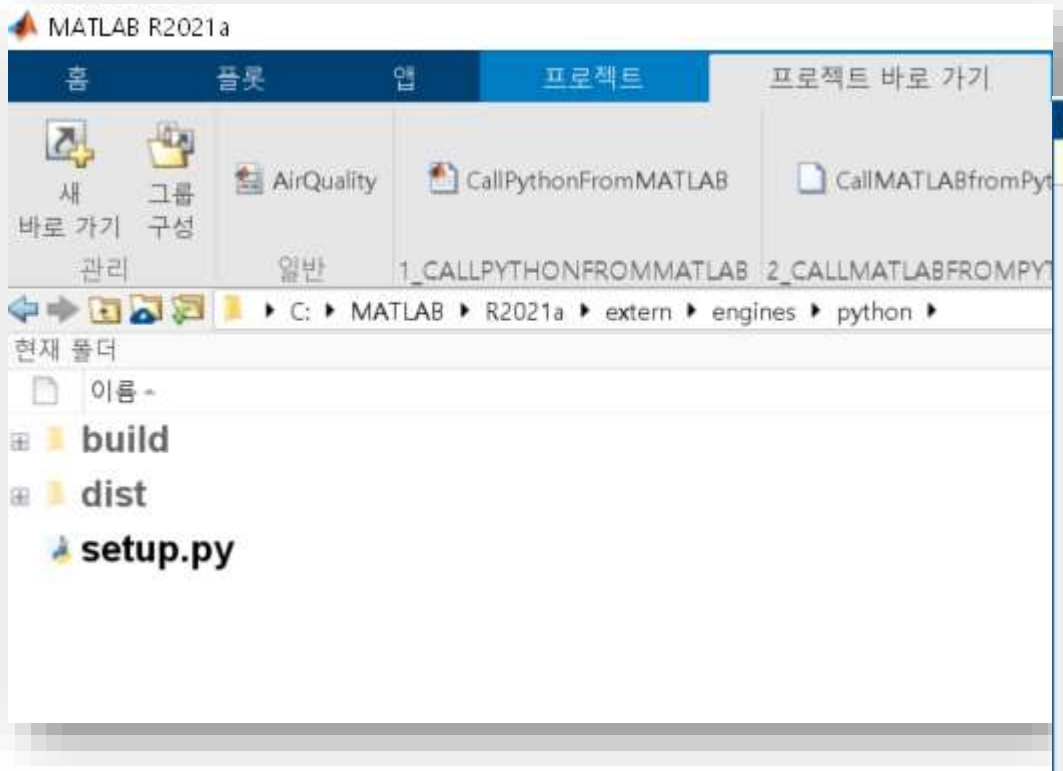
Co-Execution

- Call Python from MATLAB
- Call MATLAB from Python

Deployment



Connect to Python

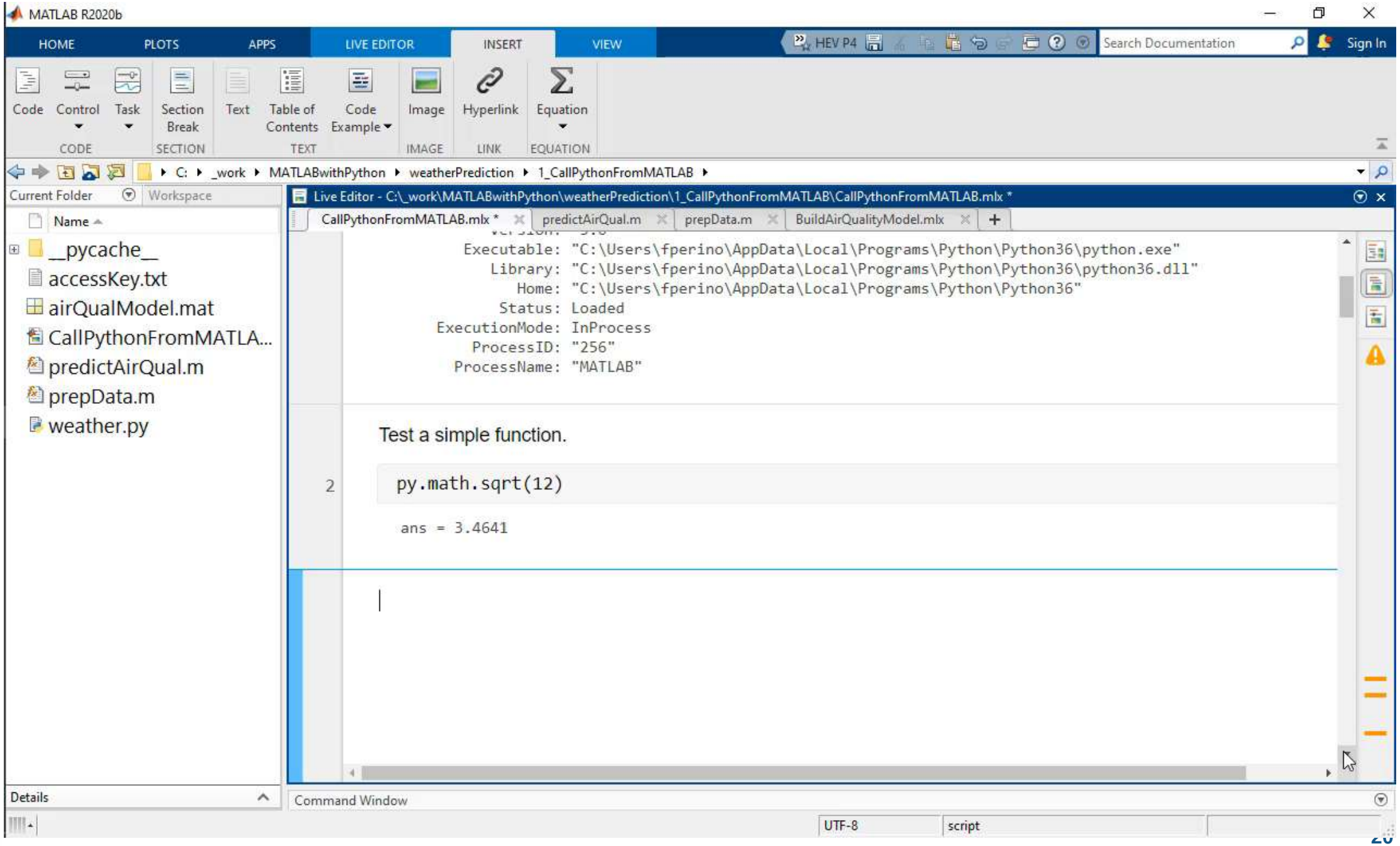


Data Access

Co-Execution

- Call Python from MATLAB
- Call MATLAB from Python

Deployment



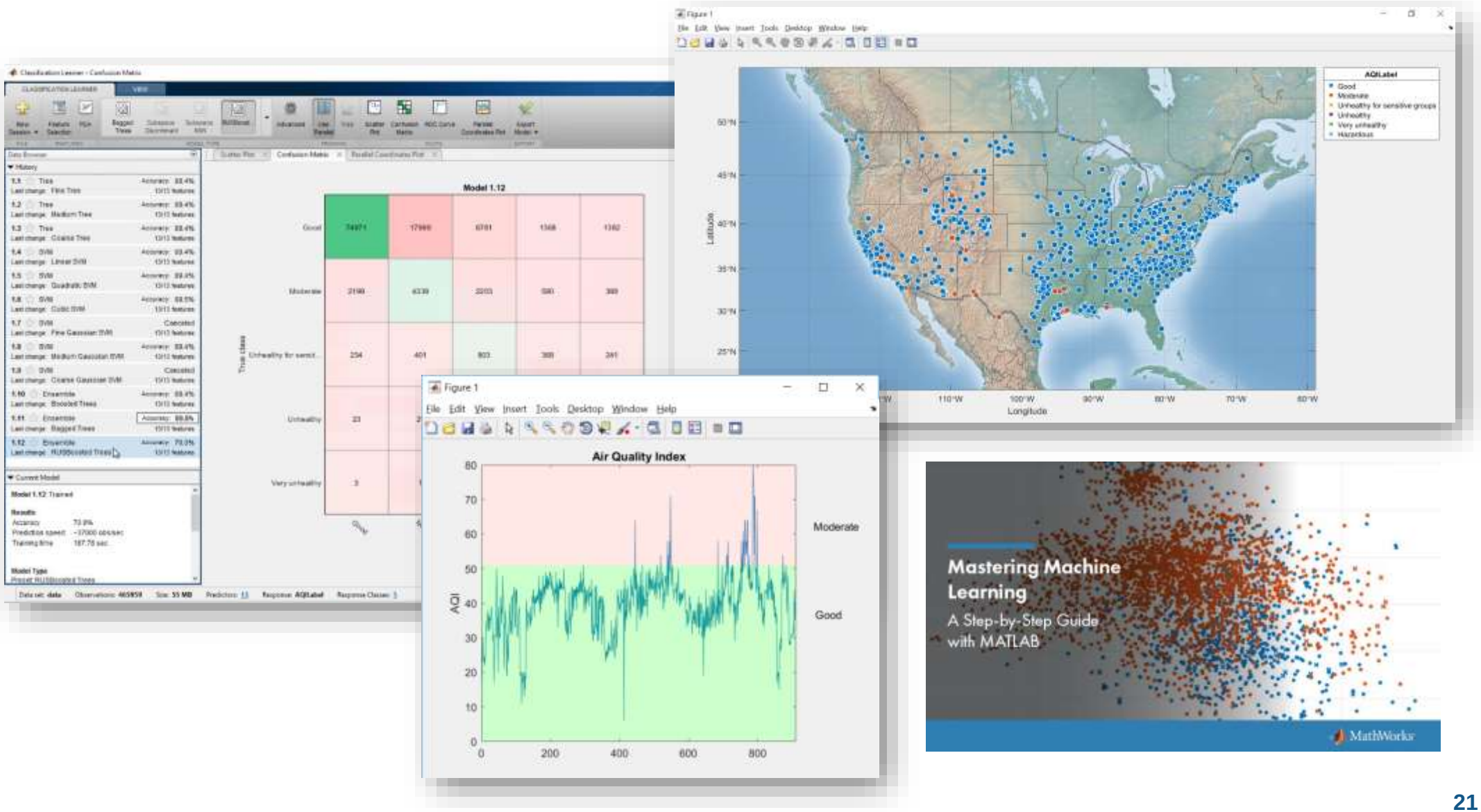
Air Quality Prediction Model

Data Access

Co-Execution

- Call Python from MATLAB
- Call MATLAB from Python

Deployment



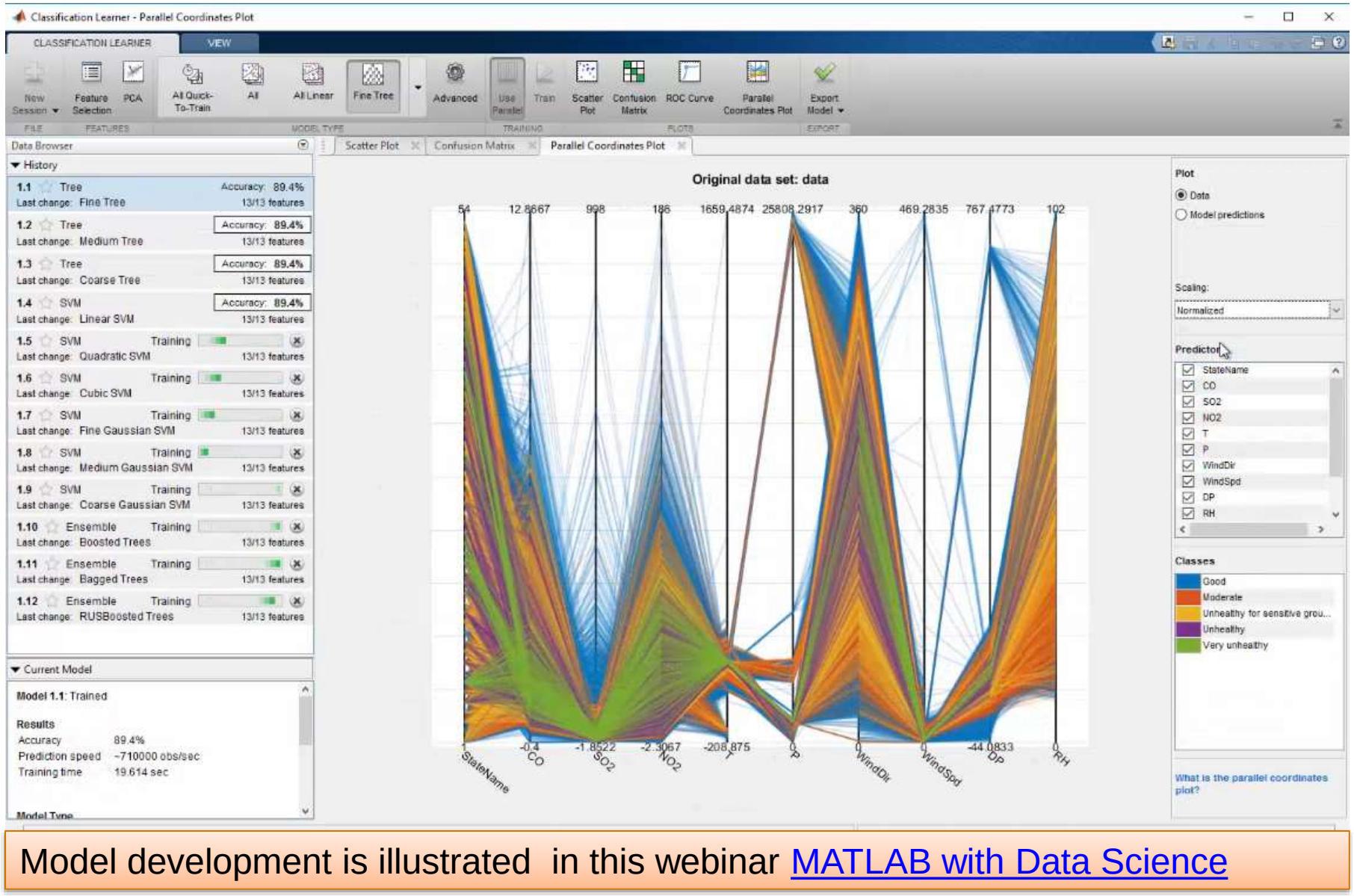
Train the Air Quality Prediction Model

Data Access

Co-Execution

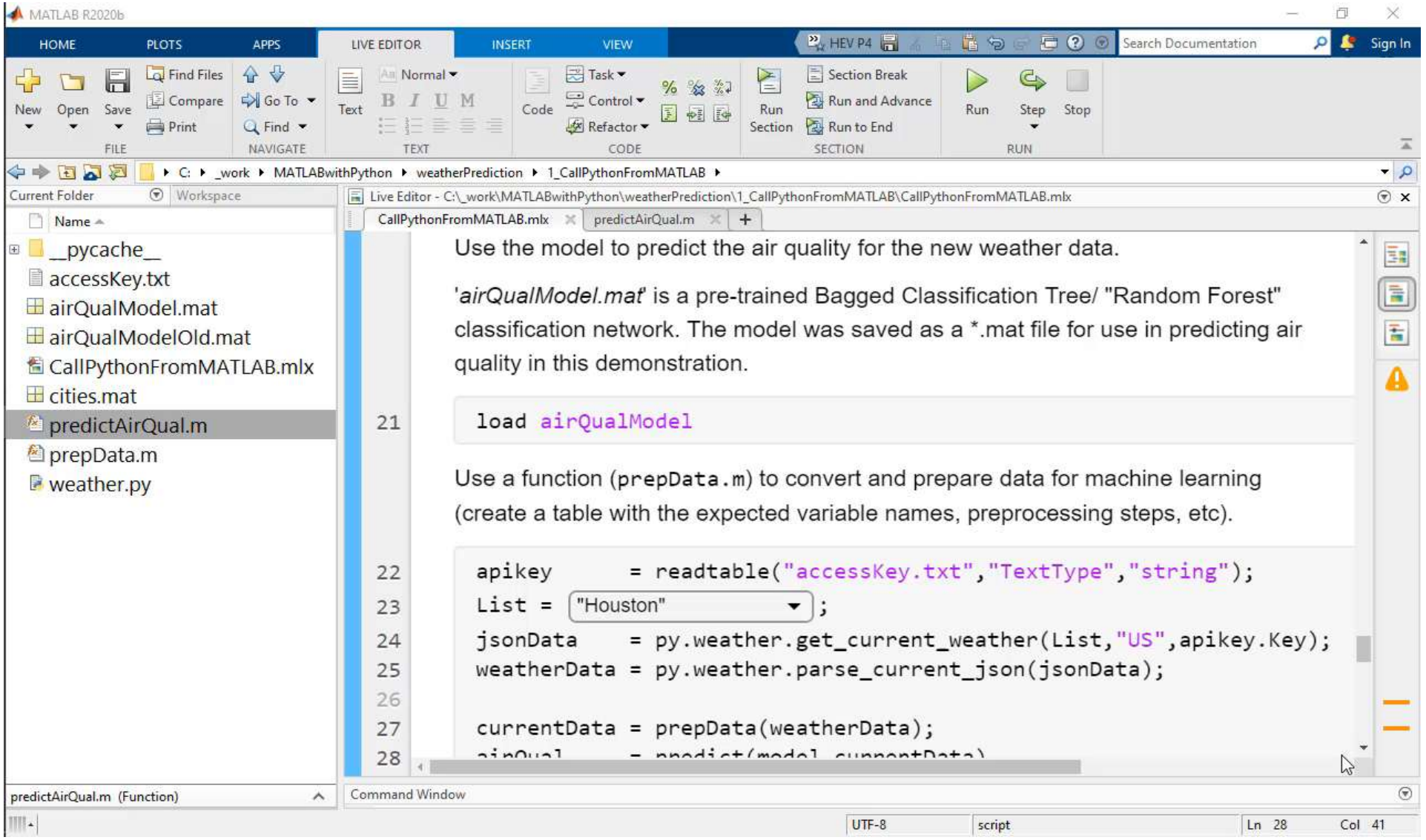
- Call Python from MATLAB
- Call MATLAB from Python

Deployment



Model development is illustrated in this webinar [MATLAB with Data Science]

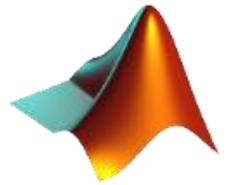
Call the Air Quality Prediction Model



Recap: Calling Python from MATLAB

Syntax differences when calling Python from MATLAB

Python



MATLAB

```
>>> import math
>>> math.sqrt(42)
```

```
>> py.math.sqrt(42)
```

```
>>> print('hello','world',sep=', ')
```

```
>> py.print('hello','world',...
       pyargs('sep',', '))
```

Data are automatically converted where possible

Otherwise convert explicitly

MATLAB to Python Data Type Mapping

R2021a

When calling a Python® function, MATLAB® converts MATLAB data into types that best represent the data to the Python language.

Pass Scalar Values to Python

MATLAB Input Argument Type — Scalar Values Only	Resulting Python `py.` Type	Examples
`double` `single`	`float`	Use Python Numeric Variables in MATLAB

Complex `single`
Complex `double`
	`complex`	`z = complex(1,2);`
		`py.cmath.polar(z)`
		
		`ans =`
		`Python tuple with no properties.`
		
		`(2.23606797749979, 1.1071487177940904)`

`int8`
`uint8`
`int16`
`uint16`
`int32`
	`int`	

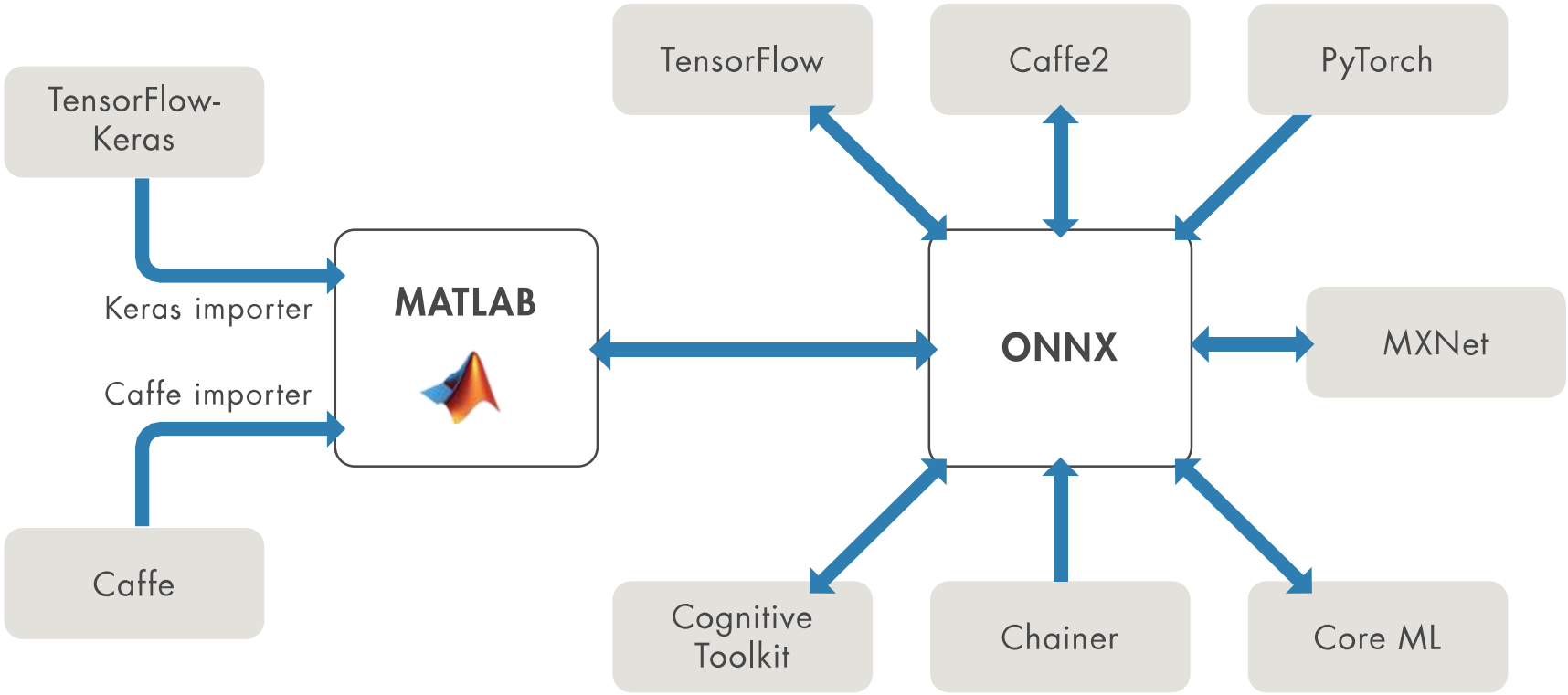
`uint32`
`int64`
`uint64`
	`int`
	`long` (version 2.7 only)	

`NaN`	`float("nan")`	
`Inf`	`float("inf")`	
`string` scalar	`str`	Use Python str Variables in MATLAB

https://mathworks.com/help/matlab/matlab_external/passing-data-to-python.html

Model Interoperability

ONNX



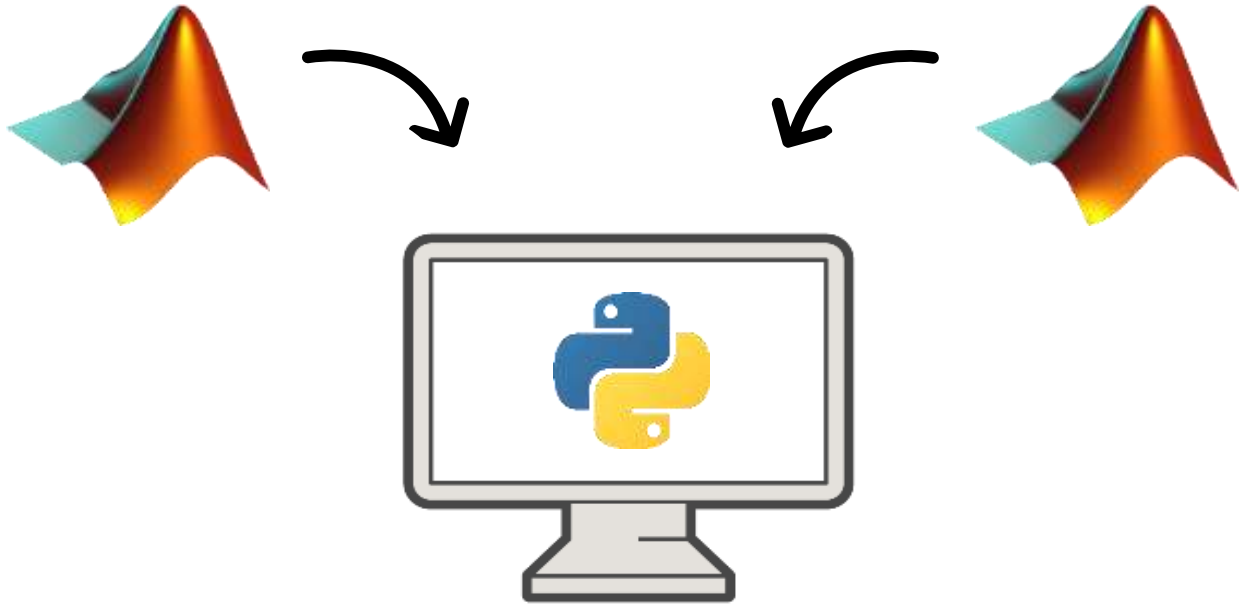
https://www.mathworks.com/solutions/deep-learning/models.html

Call MATLAB from Python

Why call MATLAB from Python?

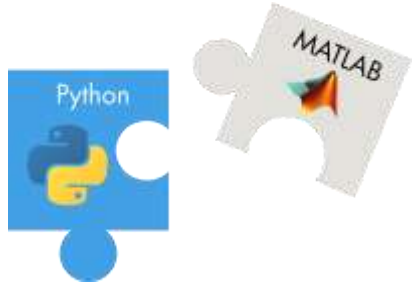
Already working in Python, and:

- Want to reuse existing MATLAB code
- Need functionality available in MATLAB
- Want to collaborate with MATLAB users



Call MATLAB from Python

To perform advanced analytics



- Calling MATLAB from Python
  - via MATLAB Engine API

```
cd (fullfile(matlabroot,'extern','engines','python'))
system('python setup.py install')
```

1. Start a MATLAB process

[1]

```
import matlab.engine
eng = matlab.engine.start_matlab()
```

2. Call MATLAB functions

[2]

```
x = eng.sqrt(float(43))
print(x)
```

```
6.557438524302
```

Integrate the MATLAB model in a Python environment

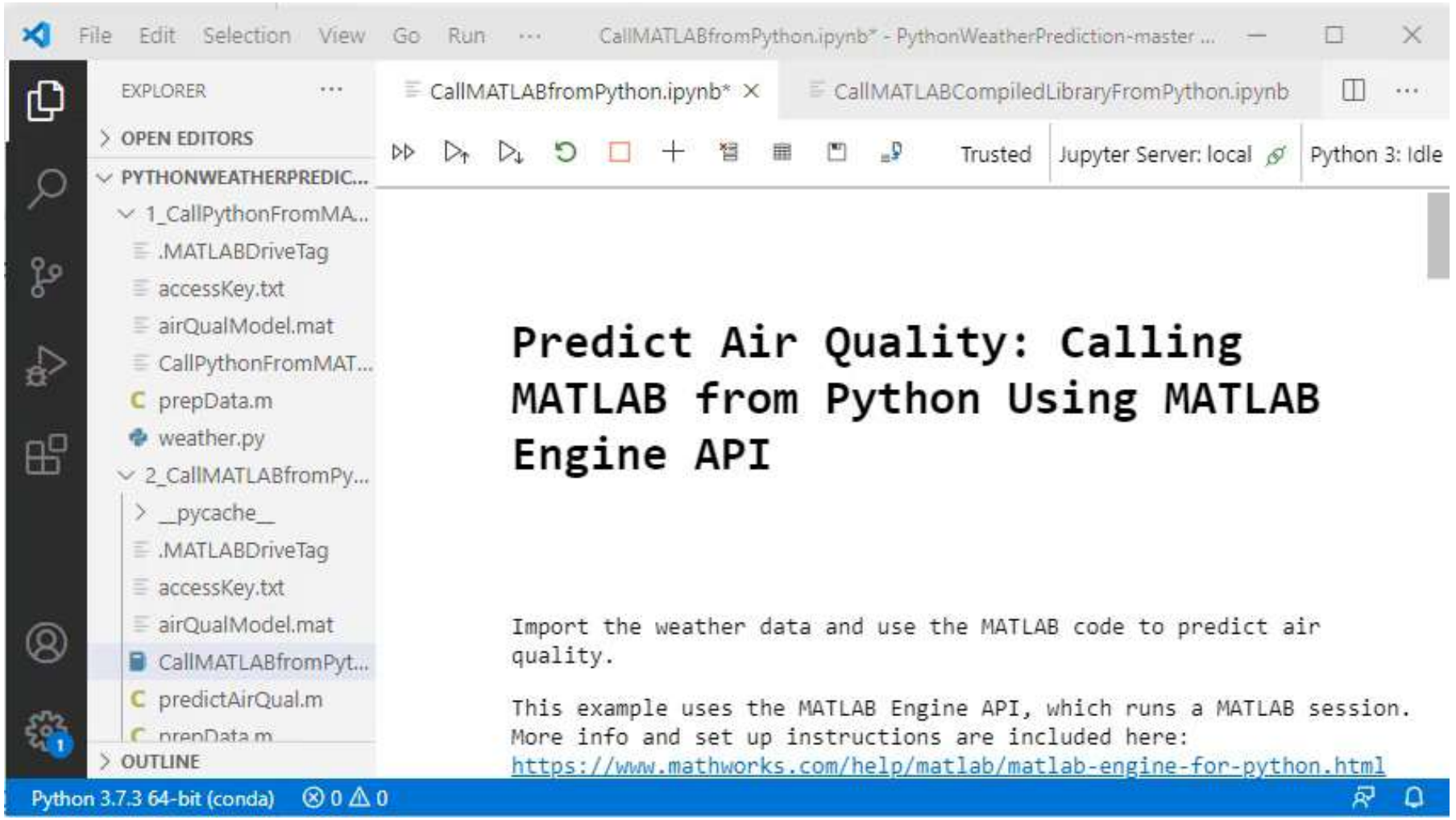
CallMATLABfromPython.ipynb

Data Access

Co-Execution

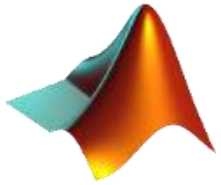
- Call Python from MATLAB
- **Call MATLAB from Python**

Deployment



Recap: Calling MATLAB from Python

Syntax differences when calling MATLAB from Python



MATLAB

Python

```
>> [s,sidx] = sort(x)
```

```
>>> s = eng.sort(x,nargout=2)
```

```
>> foo(x)
```

```
>>> eng.foo(x,nargout=0)
```

```
>> C = A + B
```

```
>>> C = eng.plus(A,B)
```

Data are automatically converted where possible

Pass Data to MATLAB from Python

R2021a

Python Type to MATLAB Scalar Type Mapping

When you pass Python® data as input arguments to MATLAB® functions, the MATLAB Engine for Python converts the data into equivalent MATLAB data types.

Python Input Argument Type — Scalar Values Only	Resulting MATLAB Data Type
`float`	`double`
`complex`	Complex `double`
`int`	`int64`
`long` (Python 2.7 only)	`int64`
`float(nan)`	`NaN`
`float(inf)`	`Inf`
`bool`	`logical`
`str`	`char`
`unicode` (Python 2.7 only)	`char`
`dict`	Structure if all keys are strings not supported otherwise

Python Container to MATLAB Array Type Mapping

Python Input Argument Type — Container	Resulting MATLAB Data Type
`matlab` numeric array object (see MATLAB Arrays as Python Variables)	Numeric array
`bytearray`	`uint8` array
`bytes` (Python 3.x) `bytes` (Python 2.7)	`uint8` array `char` array
`list`	Cell array
`set`	Cell array
`tuple`	Cell array

https://mathworks.com/help/matlab/matlab_external/pass-data-to-matlab-from-python.html

Deployment

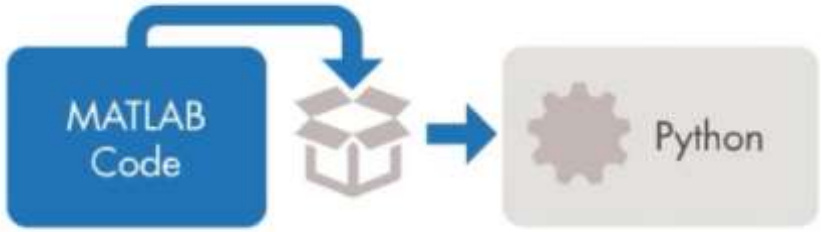
Generate Python library from MATLAB functions

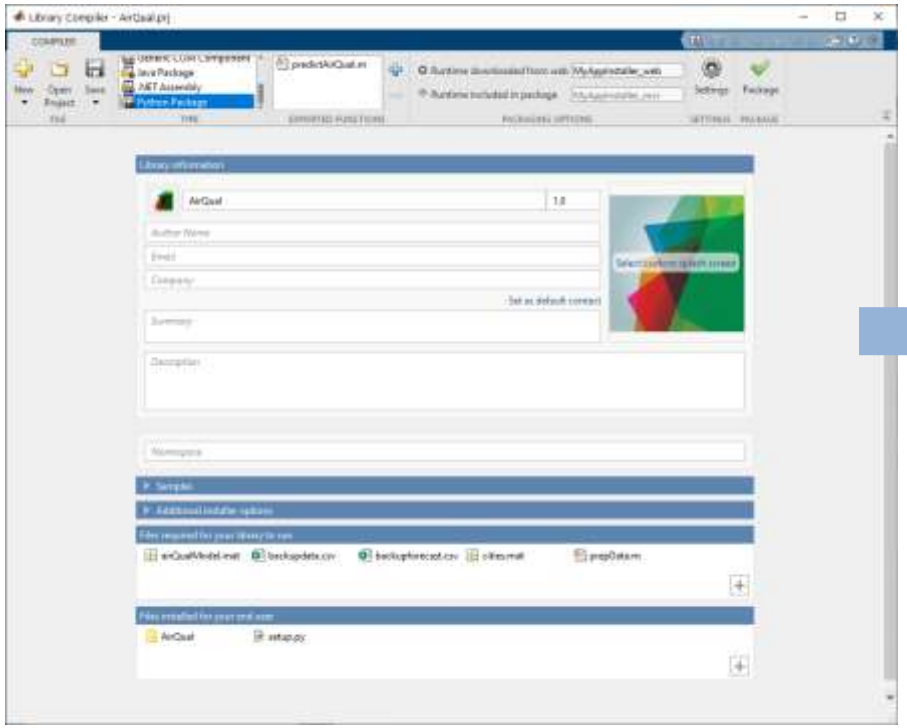
Data Access

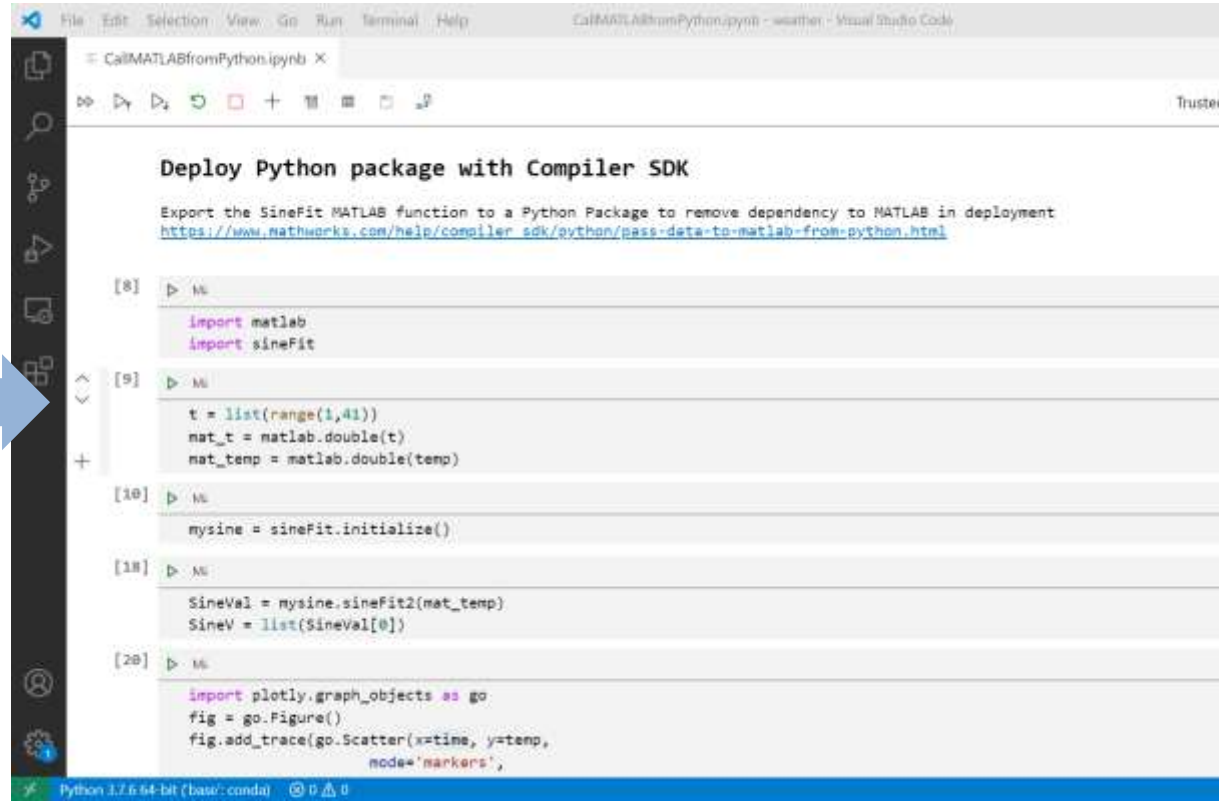
Co-Execution

- Call Python from MATLAB
- Call MATLAB from Python

Deployment







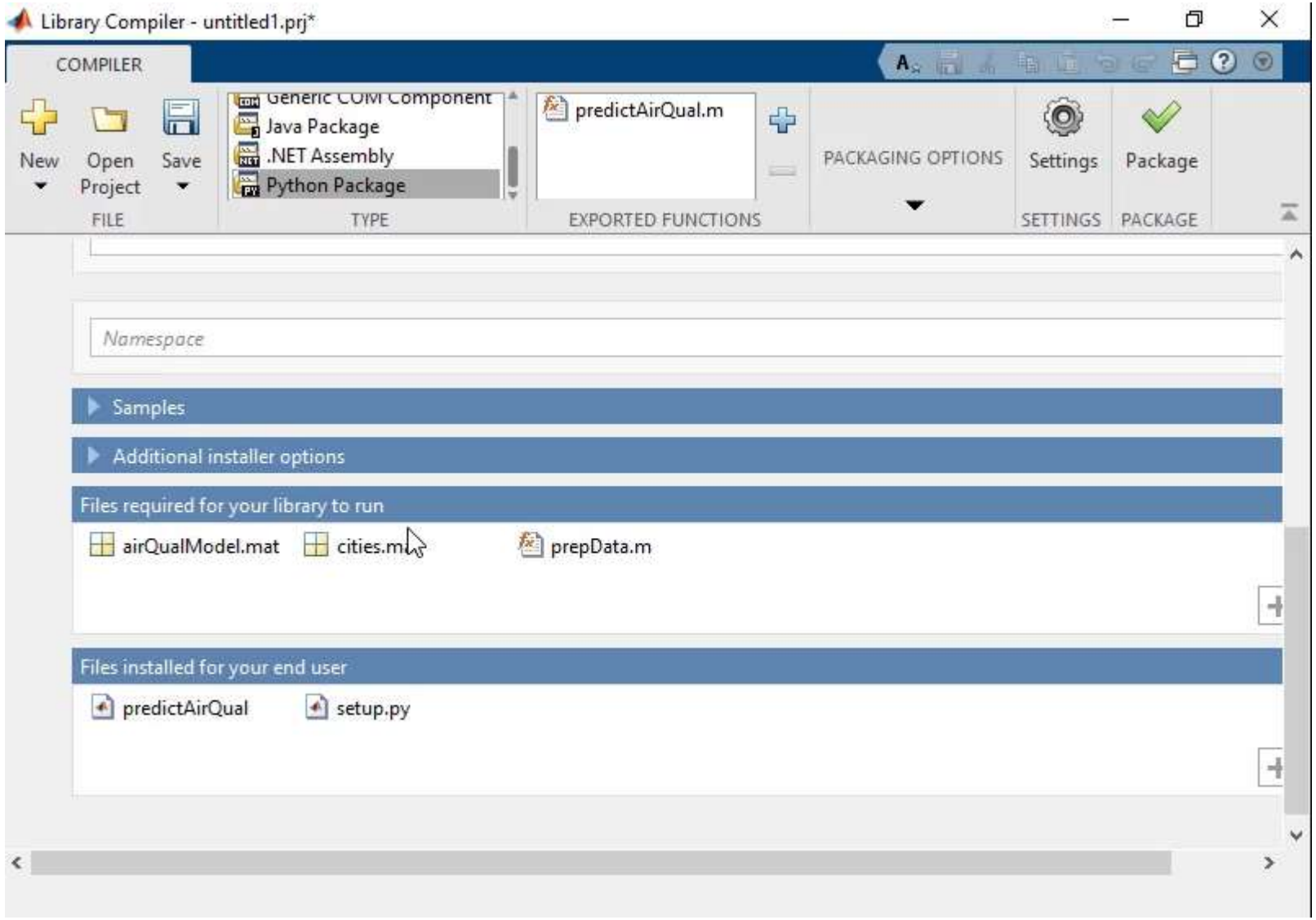
Generate Python library from MATLAB functions

Data Access

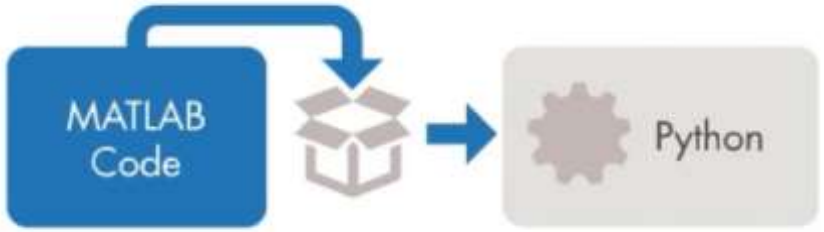
Co-Execution

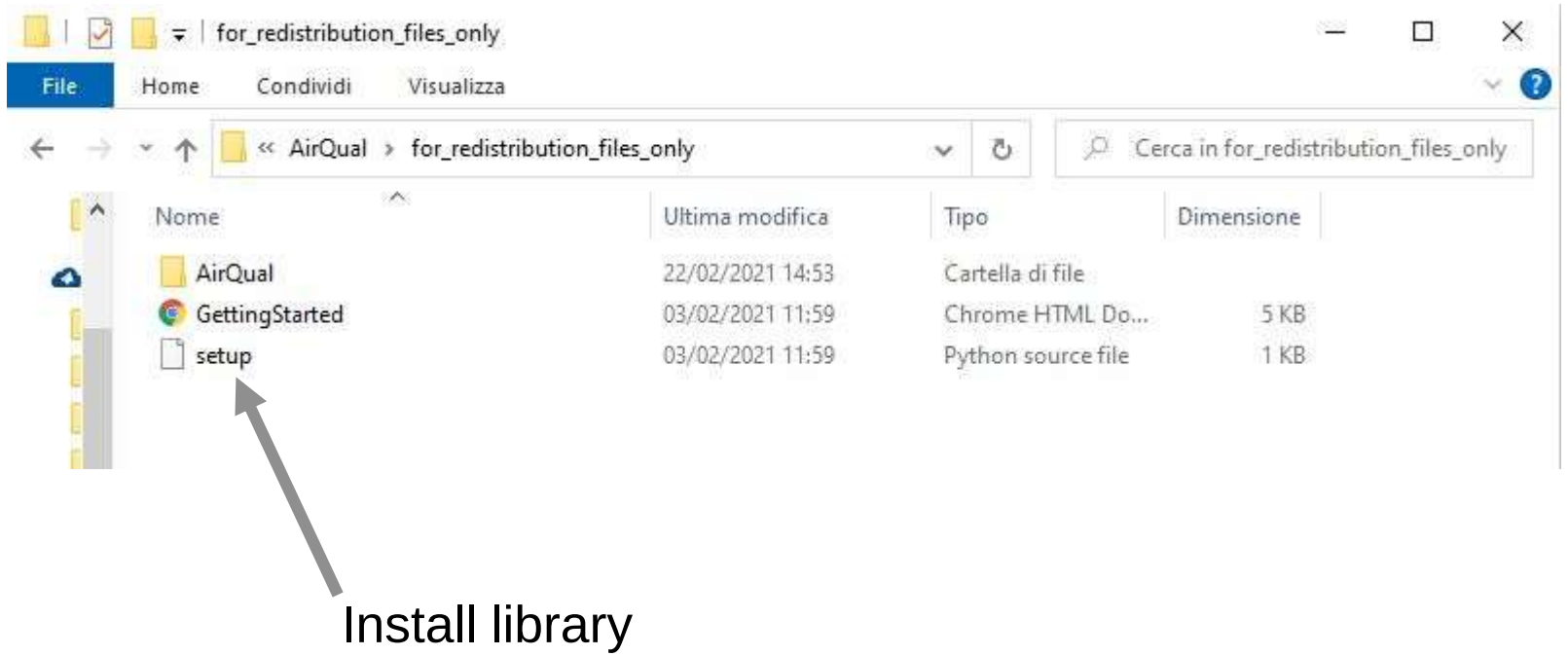
- Call Python from MATLAB
- Call MATLAB from Python

Deployment



Generate Python library from MATLAB functions





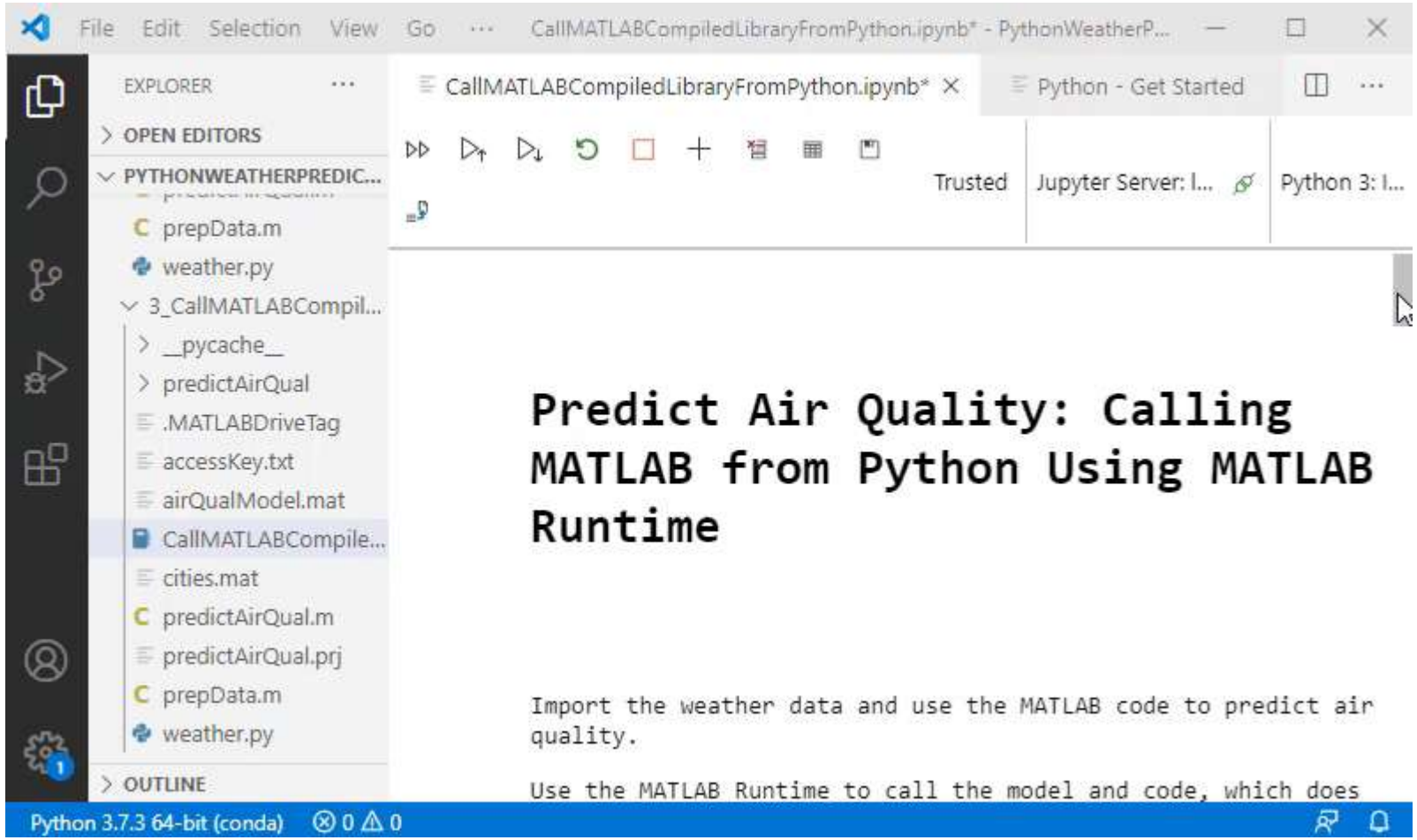
Execute Python library from MATLAB functions

Data Access

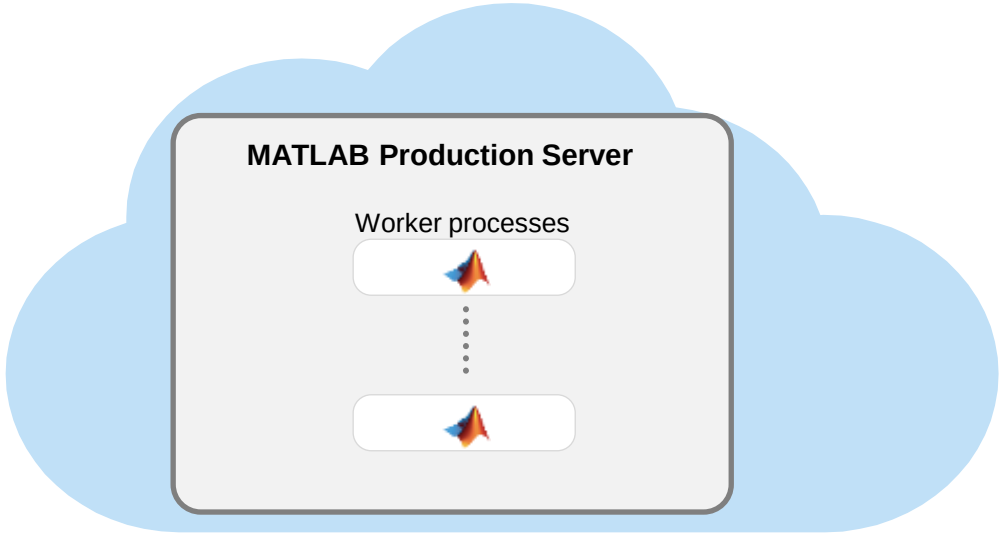
Co-Execution

- Call Python from MATLAB
- Call MATLAB from Python

Deployment



MATLAB Production Server
Access functions as web services



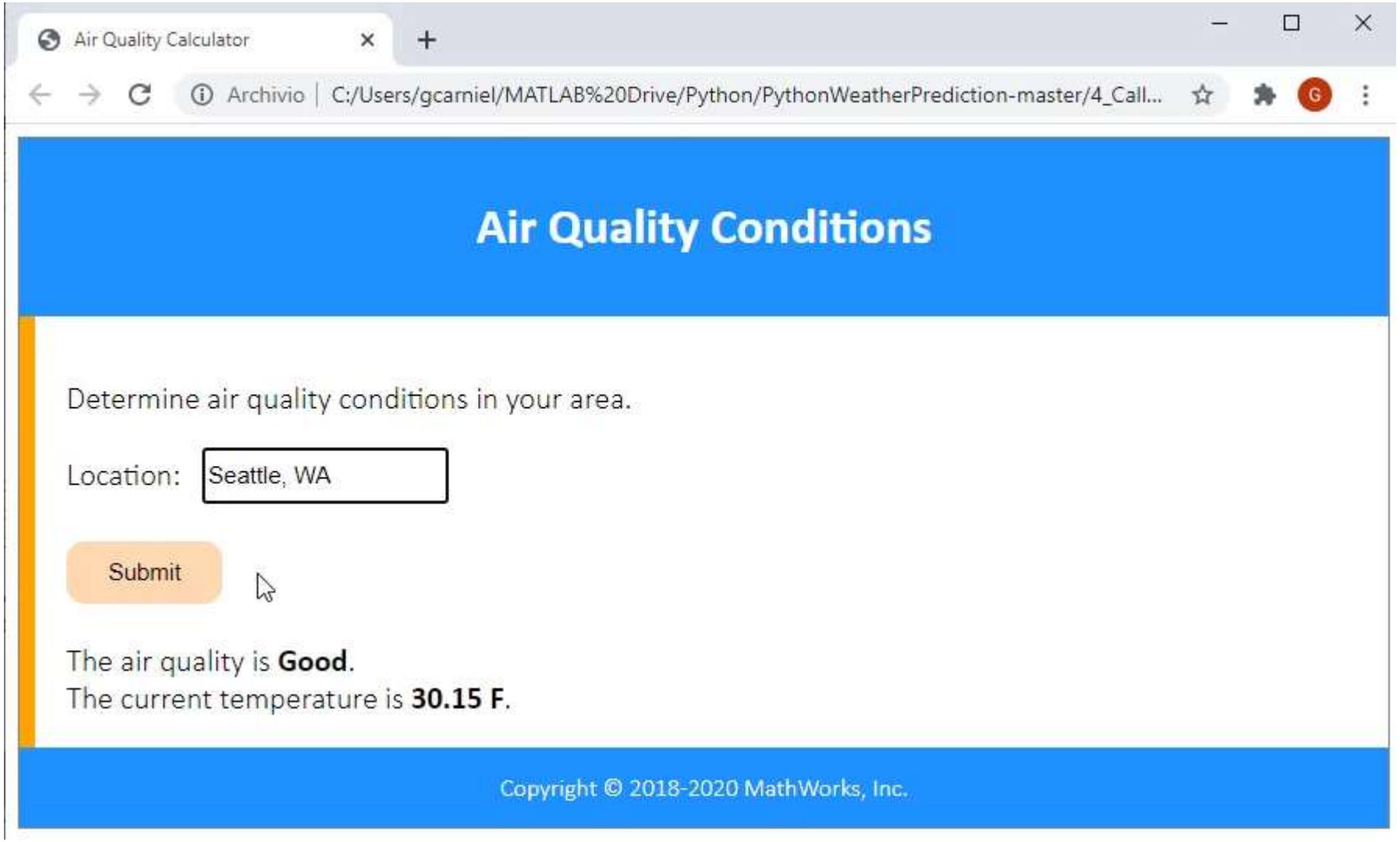
Calling our function:

```
{"nargout":1,"rhs":["input"]}
```

Getting the result:

```
{"lhs":[{"mwdata":["output"],"mwsize":[1,6],"mwtype":"char"}]}
```

Execute Python library from MATLAB functions



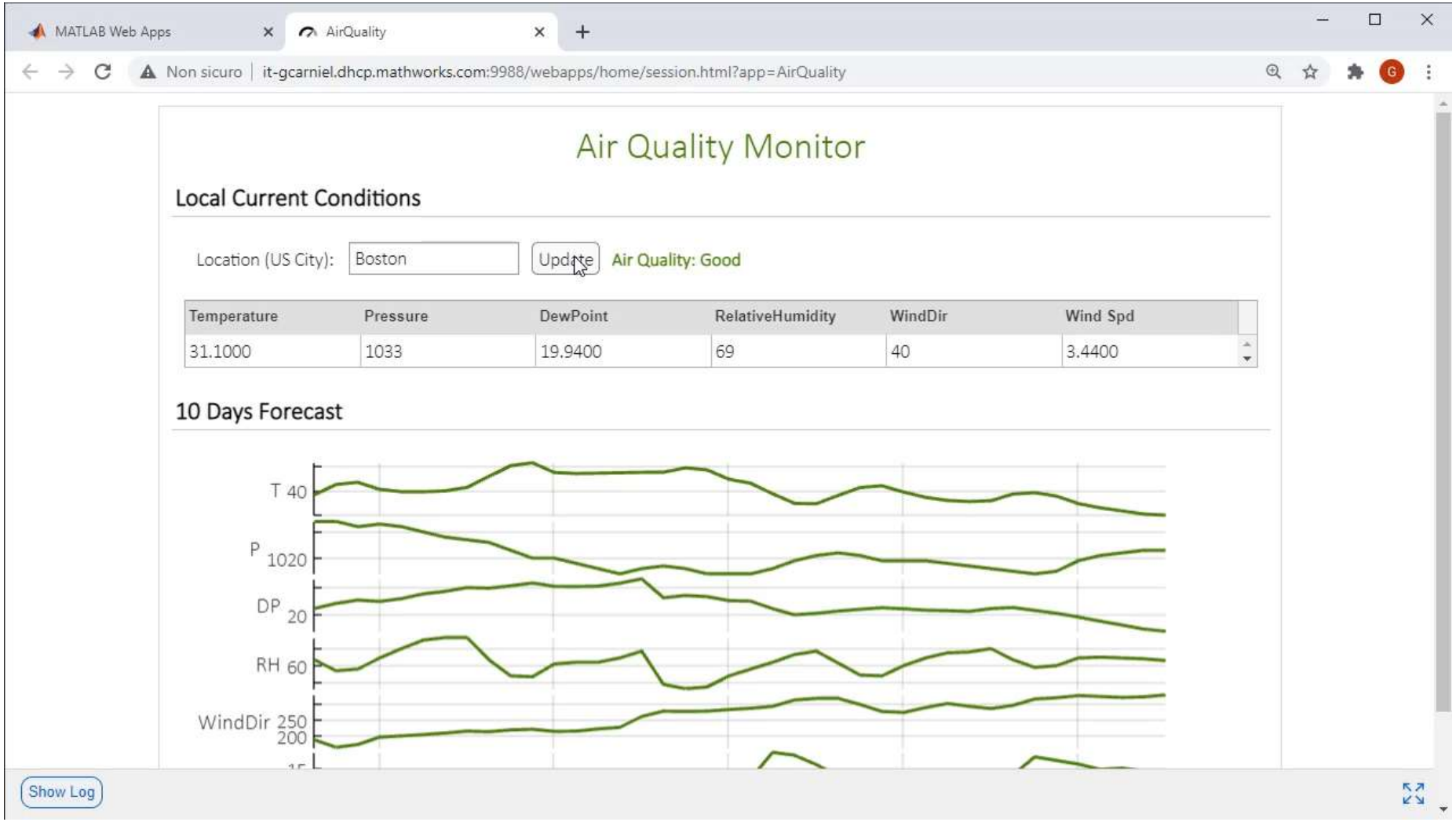
Share MATLAB App in the Web – Central Deployment

Data Access

Co-Execution

- Call Python from MATLAB
- Call MATLAB from Python

Deployment



Data Access

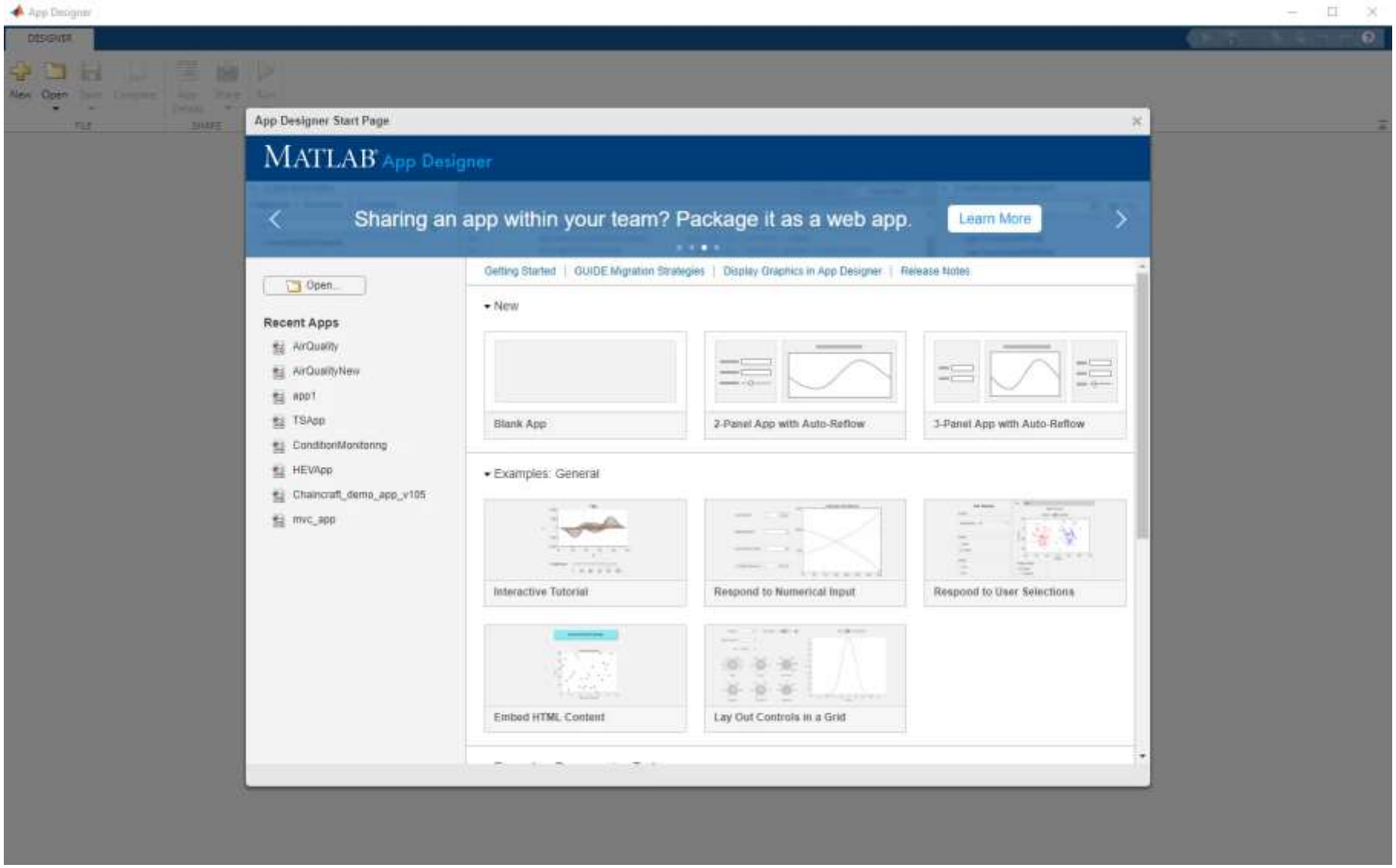
Co-Execution

- Call Python from MATLAB
- Call MATLAB from Python

Deployment

MATLAB App Designer

App development for Non-Programmers



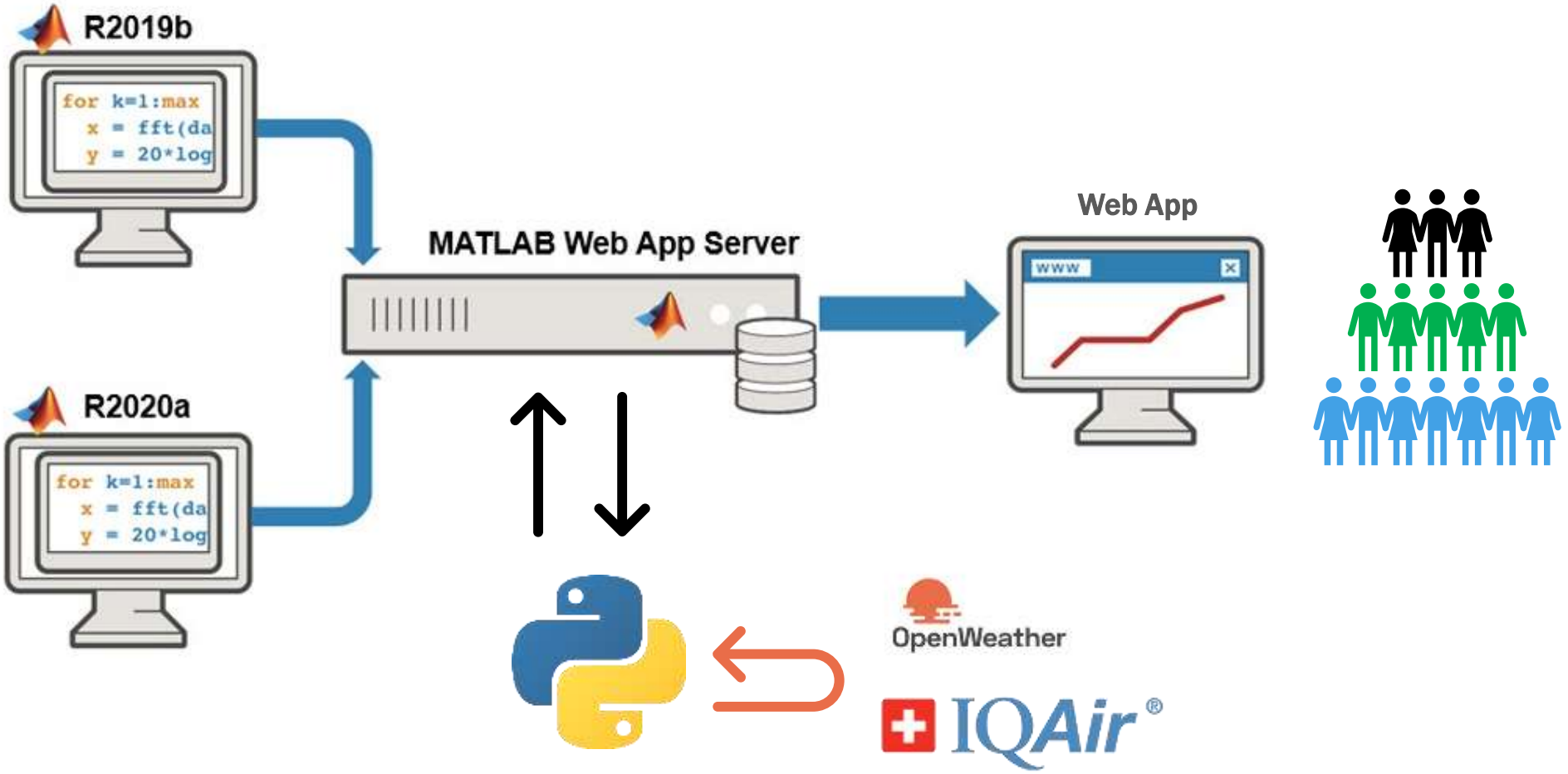
MATLAB Web App Server – Central Deployment

Data Access

Co-Execution

- Call Python from MATLAB
- Call MATLAB from Python

Deployment



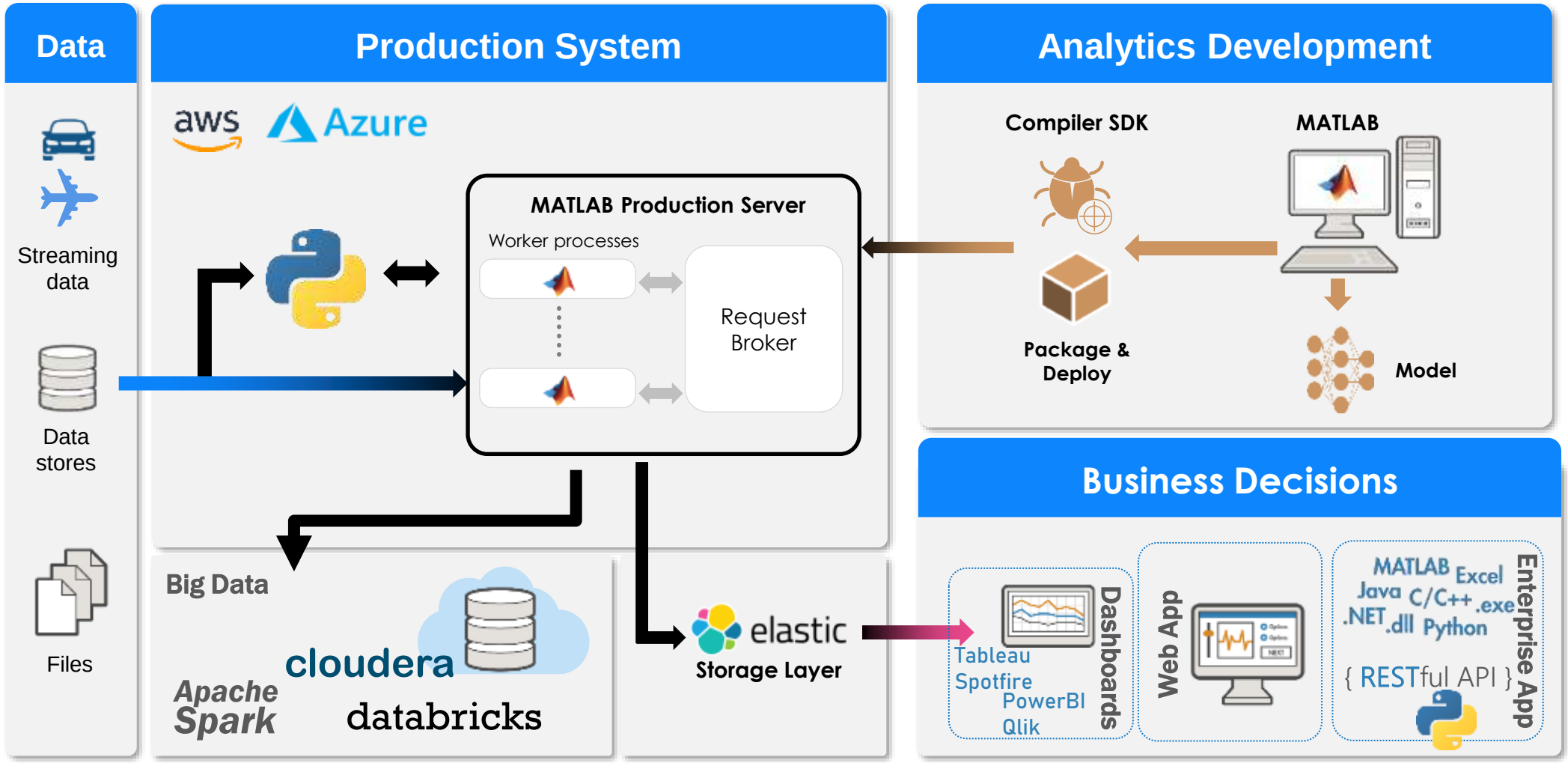
Integrate your Production System in an IT ecosystem

Data Access

Co-Execution

- Call Python from MATLAB
- Call MATLAB from Python

Deployment



Data Access

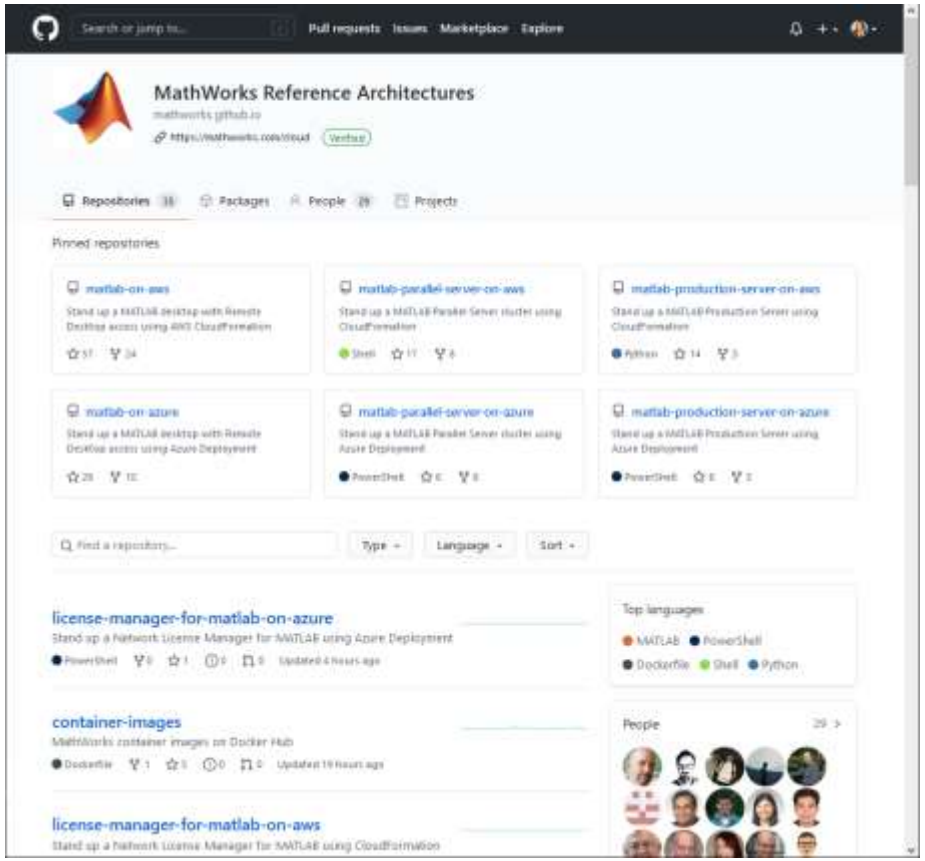
Co-Execution

- Call Python from MATLAB
- Call MATLAB from Python

Deployment

Use MATLAB Reference Architectures for easy cloud setup, Dockerfiles, and interfaces to OSS

- https://github.com/mathworks-ref-arch/matlab-dockerfile



https://github.com/mathworks-ref-arch

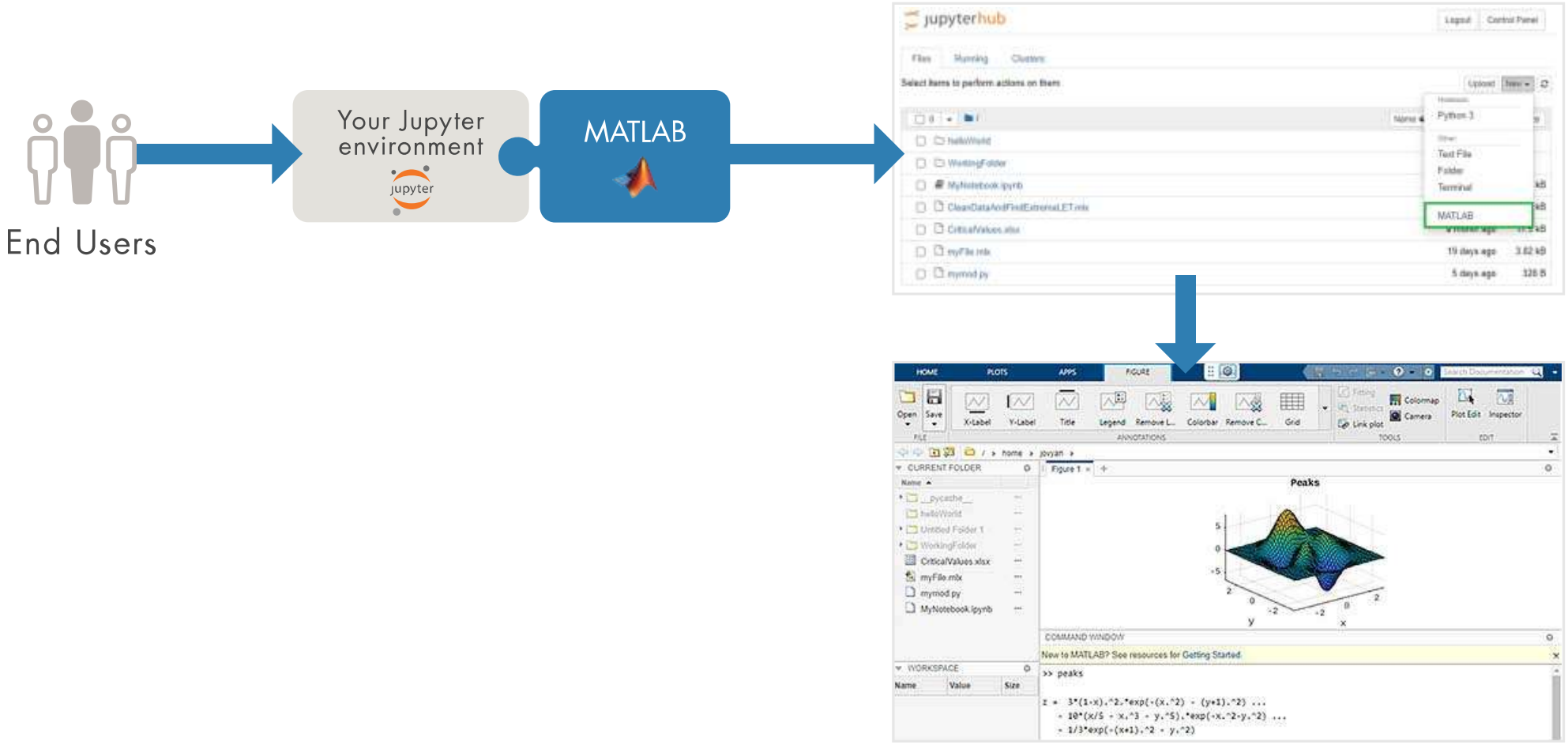
MATLAB Integration for Jupyter

Data Access

Co-Execution

- Call Python from MATLAB
- Call MATLAB from Python

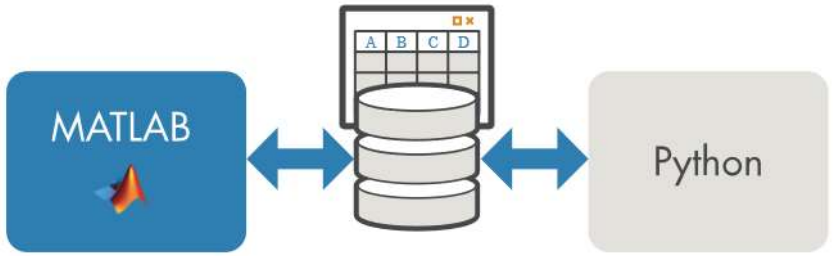
Deployment



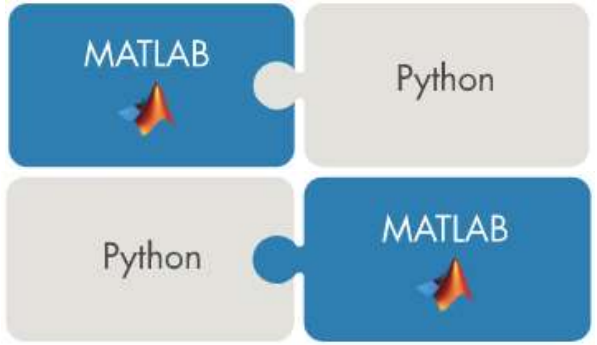
[MATLAB Integration for Jupyter (mathworks.com)]

Summary:
Using MATLAB with Python

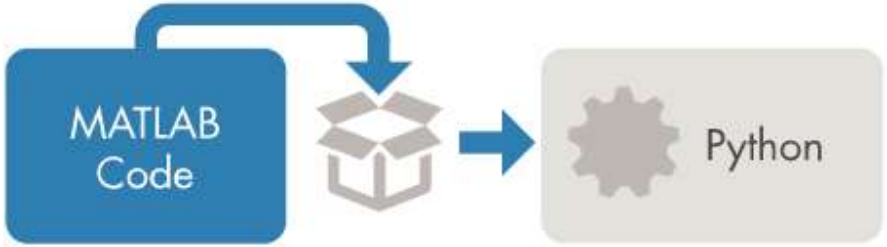
- Access Data
  - Weather App example



- Interoperability
  - Calling libraries written in Python from MATLAB
  - Calling MATLAB from Python



- Deploy Apps & Algos
  - Web App
  - Production API



Additional resources

Resources

- General:
  - https://www.mathworks.com/products/matlab/matlab-and-python.html
- Python from MATLAB:
  - https://www.mathworks.com/help/matlab/call-python-libraries.html
- MATLAB from Python:
  - MATLAB Engine API:
    - https://www.mathworks.com/help/matlab/matlab-engine-for-python.html
  - MATLAB Compiler SDK:
    - https://www.mathworks.com/help/compiler_sdk/python_packages.html
  - Data type conversions:
    - https://www.mathworks.com/help/matlab/python-data-types.html
- Example:
  - https://github.com/mathworks/matlab-with-python

Cheatsheet

MathWorks
Accelerating the pace of engineering and science

Using MATLAB® and Python® Together

The ≥ icon provides links to relevant sections of the MATLAB documentation to learn more.

Call Python in MATLAB

Access settings and status of Python interpreter:

```
>> pe = pyenv
```

Specify version to use:

```
>> pe = pyenv("Version",3.7)
```

Call Python modules and functions:

```
py.module_name.function_name
```

```
>> py.math.sqrt(42)
```

Pass keyword arguments

Use pyargs to pass keyword arguments

```
>>> foo(5,bar=42)
>> py.foo(5,pyargs('bar',42))
```

Reload modules

Reload the module after making updates:

```
>> py.importlib.reload(module)
```

Call MATLAB in Python

Install MATLAB Engine API for Python ≥

Run `setup.py` from an OS command window

```
$ cd [matlabroot]/extern/engines/python
$ python setup.py install
```

Call MATLAB functions

Import the module and start the engine

```
>>> import matlab.engine
>>> eng =
    matlab.engine.start_matlab()
```

Call functions through the engine

```
>>> x = eng.sqrt(42.0)
```

Capture multiple outputs

```
>>> x = eng.gcd(42.0,8.0,nargout=3)
```

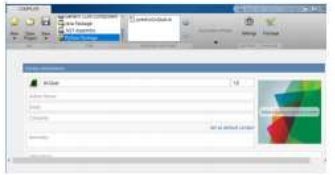
Stop the engine

```
>>> eng.exit()
```

Create Python Package

Package MATLAB functions ≥

Use the Library Compiler App to create a Python package for MATLAB functions



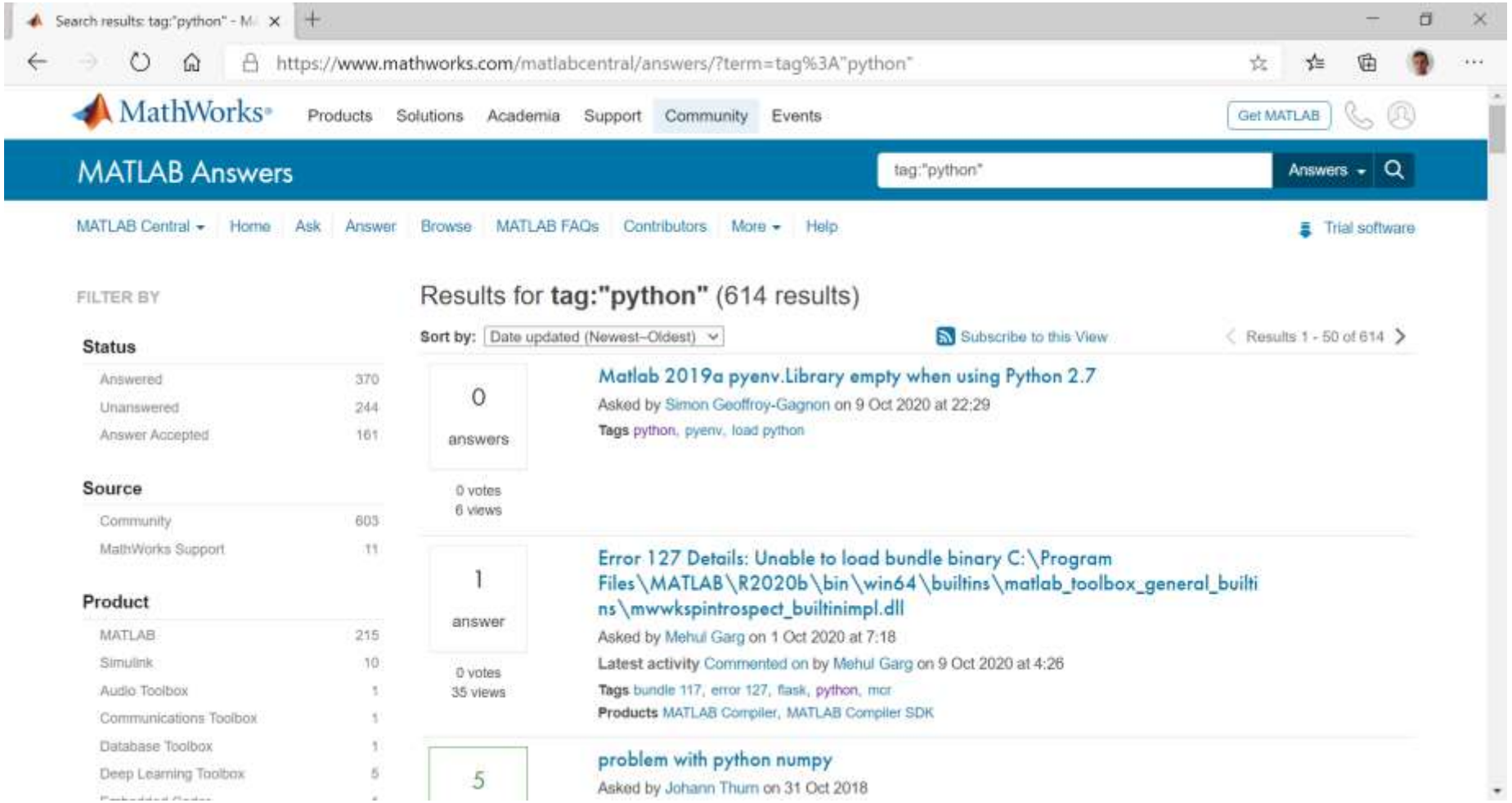
Invoke MATLAB functions from the Python package

```
>>> import PackageName
>>> pkg =
PackageName.initialize()
>>> result = pkg.foo()
```

Close package

```
>>> pkg.terminate()
```

MATLAB Answers – tag:"python"



https://www.mathworks.com/matlabcentral/answers/?term=tag%3A%22python%22

Additional Resources

- ***NEW*** Release
  - https://www.mathworks.com/products/new_products/latest_features.html
- Predictive Analytics
  - https://www.mathworks.com/discovery/predictive-analytics.html
- Deep Learning
  - https://www.mathworks.com/discovery/deep-learning.html
- Reinforcement Learning
  - https://www.mathworks.com/products/reinforcement-learning.html