

# #16

## Qiskit Metal to cloud-ready

---

Marco Facchini @marcolincs & Yehan Liu @quantum-quant

Scott Wyman Neagle @scottwn

Dayeong Kang @tula3and

The goal:  
provide  
the qiskit-metal tool  
as a service

1. Execute qiskit metal in back-end
2. Visualize the results from backend to front-end (with #15)

#### In back-end:

1. Get an input file from FE
2. Execute the input and make a result file
3. Send the result to FE

#### In front-end:

1. Make an input file with the user-interactive tool
2. Get a result file from BE and show in the webpage

The goal:

provide  
the qiskit-metal tool  
as a service

1. Execute qiskit  
metal in back-end

2. Visualize the  
results from  
back-end to front-  
end (with #1)

In back-end:

1. Get an input file from FE

2. Execute the input and  
make a result file

3. Send the result to FE

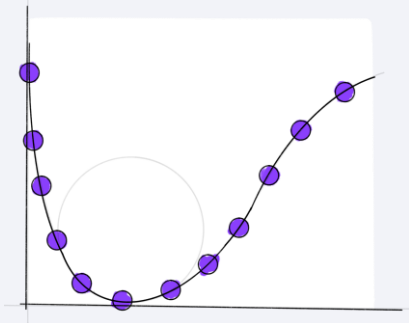
In front-end:

1. Make an input file with  
the user-interactive tool

2. Get a result file from BE  
and show in the webpage

# How to Visualize?

## How to visualize?



1.

Understanding  
how to make the command  
``import qiskit-metal``  
execute without errors  
in the IBM Quantum Lab  
notebook

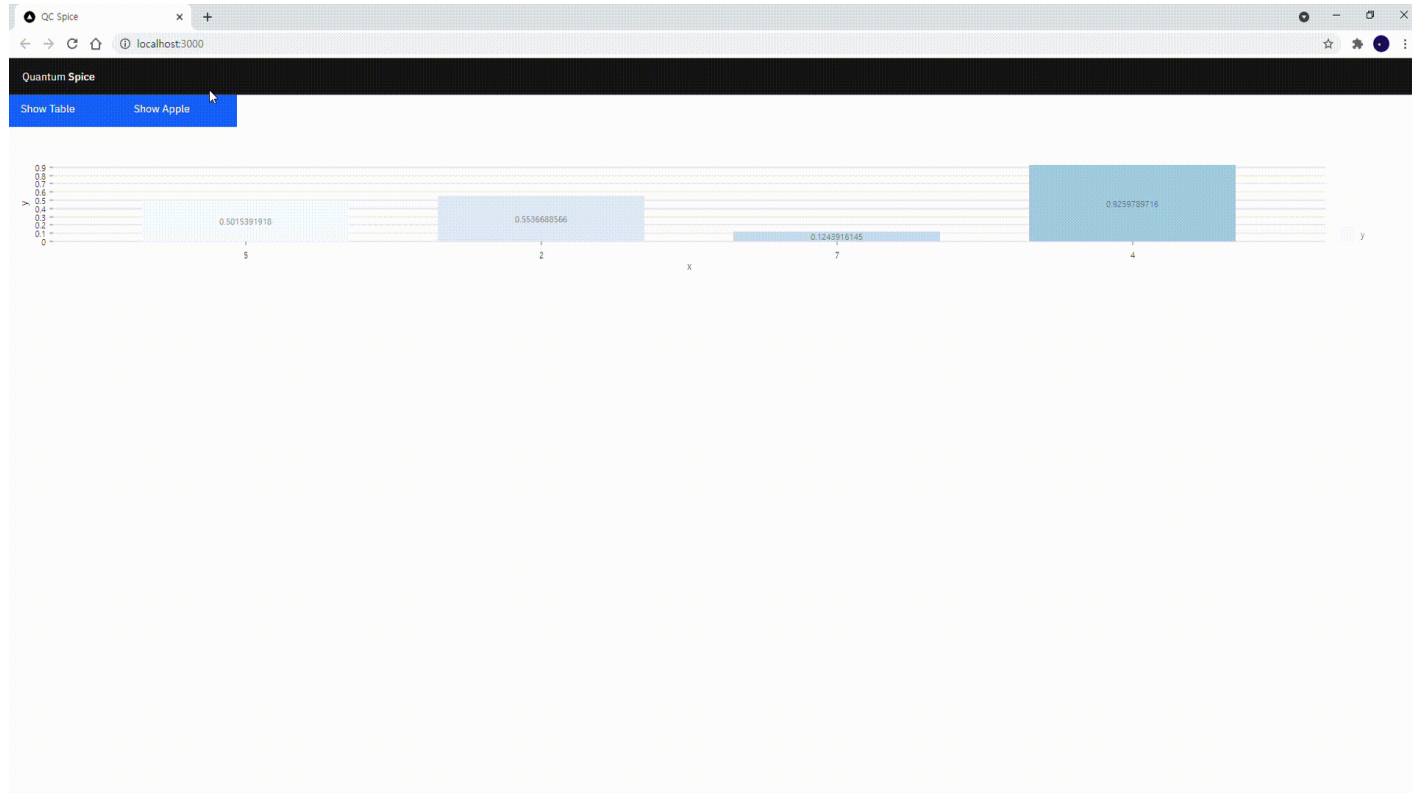
2.

Analyzing the output file &  
Making a plot for the data

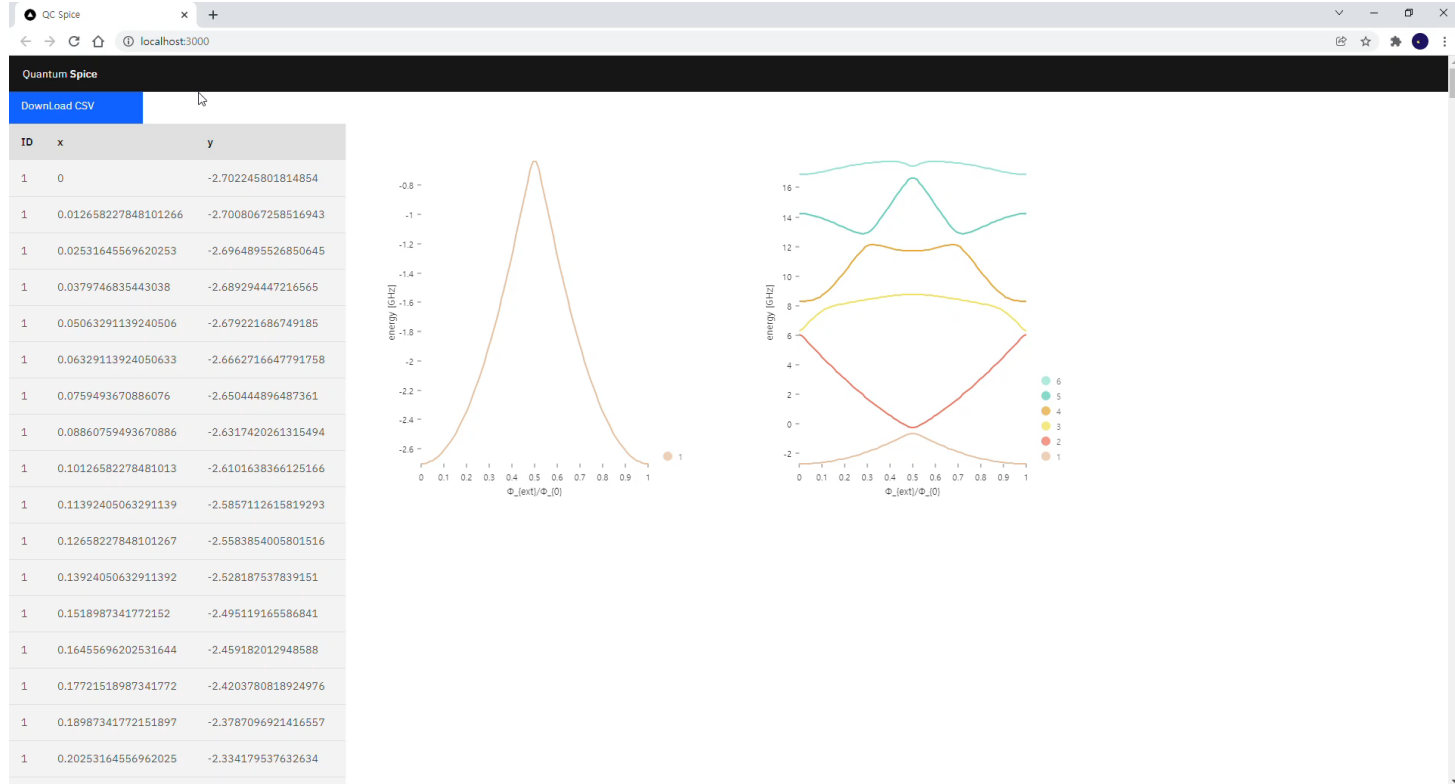
# How to execute without errors

```
tula@ : giskit-metal/qiskit_metal$ grep -r 'PySide' .
./renderers/renderer_mpl/extensions/animated_text.py:from PySide2.QtCore import QTimer
Binary file ./renderers/renderer_mpl/extensions/__pycache__/animated_text.cpython-39.pyc matches
./renderers/renderer_mpl/mpl_canvas.py:from PySide2.QtCore import QTimer
./renderers/renderer_mpl/mpl_canvas.py:from PySide2.QtWidgets import QSizePolicy
./renderers/renderer_mpl/mpl_interaction.py:from PySide2.QtCore import Qt
./renderers/renderer_mpl/mpl_interaction.py:from PySide2.QtGui import QIcon
./renderers/renderer_mpl/mpl_interaction.py:from PySide2.QtWidgets import QAction, QLabel
./renderers/renderer_mpl/mpl_interaction.py: # TODO: Remove use of tool manager just use PySide2 bare as below
Binary file ./renderers/renderer_mpl/__pycache__/mpl_canvas.cpython-39.pyc matches
Binary file ./renderers/renderer_mpl/__pycache__/mpl_interaction.cpython-39.pyc matches
./toolbox_metal/about.py: from PySide2.QtCore import __version__ as QT_VERSION_STR
./toolbox_metal/about.py: from PySide2 import __version__ as PYSIDE_VERSION_STR
./toolbox_metal/about.py: PySide2 version {PYSIDE_VERSION_STR}
./toolbox_metal/exceptions.py: """Run PySide2 only.
./toolbox_metal/exceptions.py: prefix = "You should be using be PySide2. \n"
Binary file ./toolbox_metal/__pycache__/about.cpython-39.pyc matches
Binary file ./toolbox_metal/__pycache__/exceptions.cpython-39.pyc matches
./toolbox_python/display.py: from PySide2.QtWidgets import QApplication, QMainWindow
```

# First one looks like...



# Still working on!

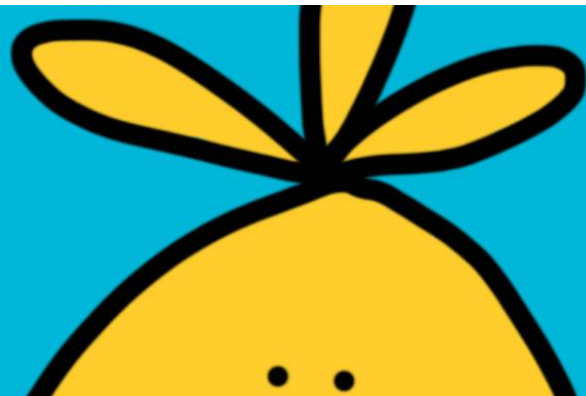


# Thank you for listening

Dayeong Kang @tula3and

## TULA Log

in the ocean of curiosity



<https://tula3and.github.io/project/project-cloud/>  
<https://tula3and.github.io/project/project-react-visualization/>