

# Cloud control for cold atoms

Presented by

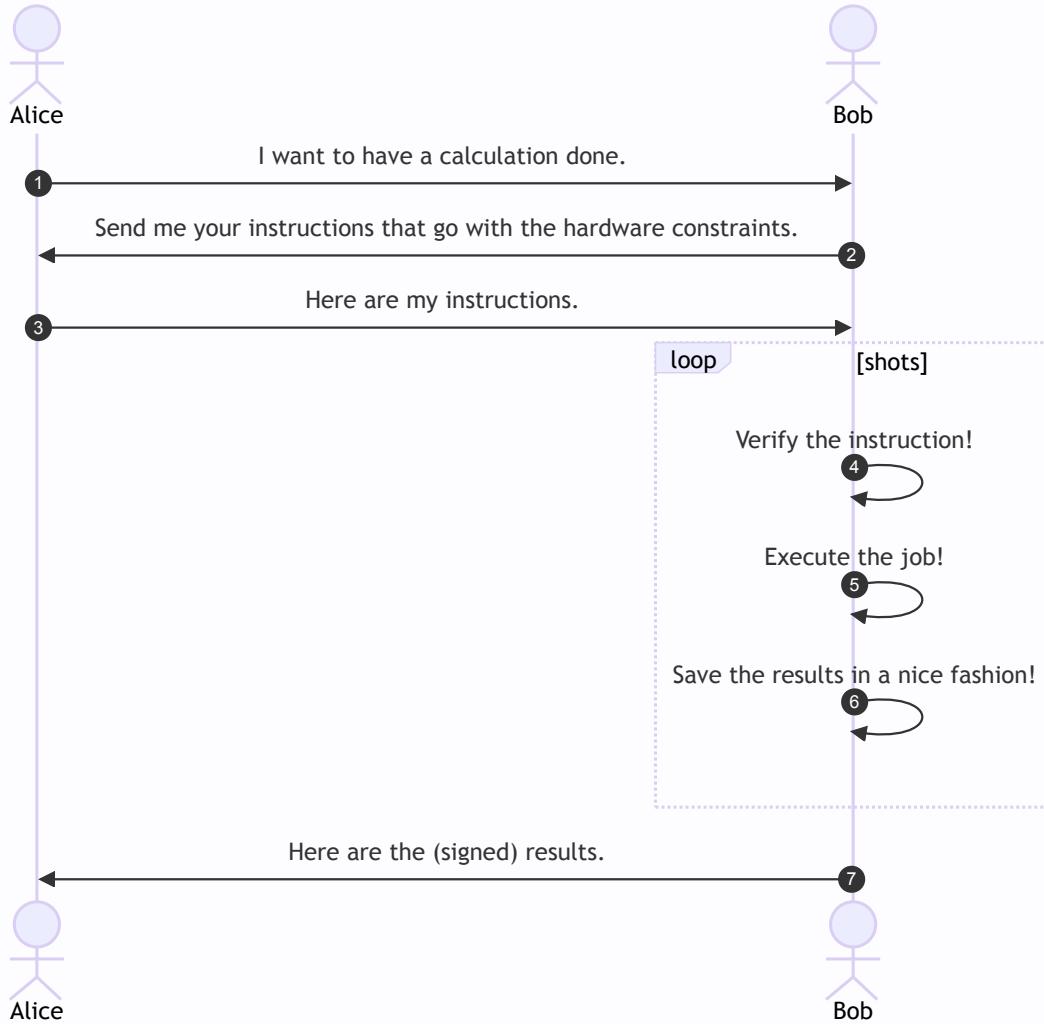
Fred J - aka [fretchen](#)



# Why cold atoms ?

Cold atoms are:

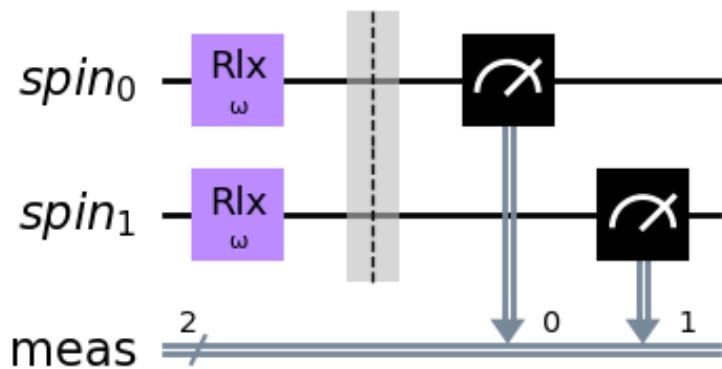
- versatile with *many* different applications
- precise and accurate
- broadly used in research and industry
- a leading quantum information platform.



# The challenge

- **Alice** is the interested user
- **Bob** is the owner of the quantum hardware.
- Now they need to communicate in an efficient way.

# Alice uses QISKit



Write the quantum circuit in  
qiskit-cold-atom.

```
Nwires = 2
qc_rabi = QuantumCircuit(QuantumRegister(Nwires, "spin"))

all_modes=range(Nwires)
omega_t = Parameter("ω")

qc_rabi.rlx(omega_t, [0, 1])

qc_rabi.measure_all()
qc_rabi.draw(output='mpl')
```

# Alice calculates locally

For small circuits Alice can simply execute the code on her own machine.

```
from qiskit_cold_atom.spins import SpinSimulator  
  
backend = SpinSimulator()  
phases = np.linspace(0, 2*np.pi, 15)  
  
rabi_list = [ qc_rabi.assign_parameters( {omega_t: phase}, inplace=False, ) for phase in phases ]  
  
n_shots = 500  
job_rabi = backend.run(rabi_list, shots=n_shots)  
result_rabi = job_rabi.result()  
counts_rabi = result_rabi.get_counts()
```

SpinSimulator is a local simulator for cold atoms and works for up to 10 spins.

# Alice chooses a cloud backend

For more complex problem Alice can send the code to a cloud backend with ColdAtomProvider .

```
from qiskit_cold_atom.providers import ColdAtomProvider  
  
cloud_backend = ColdAtomProvider()  
  
job_remote_rabi = cloud_backend.run(remote_rabi_list, shots=500)  
remote_result_rabi = job_rabi.result()  
remote_counts_rabi = result_rabi.get_counts()
```

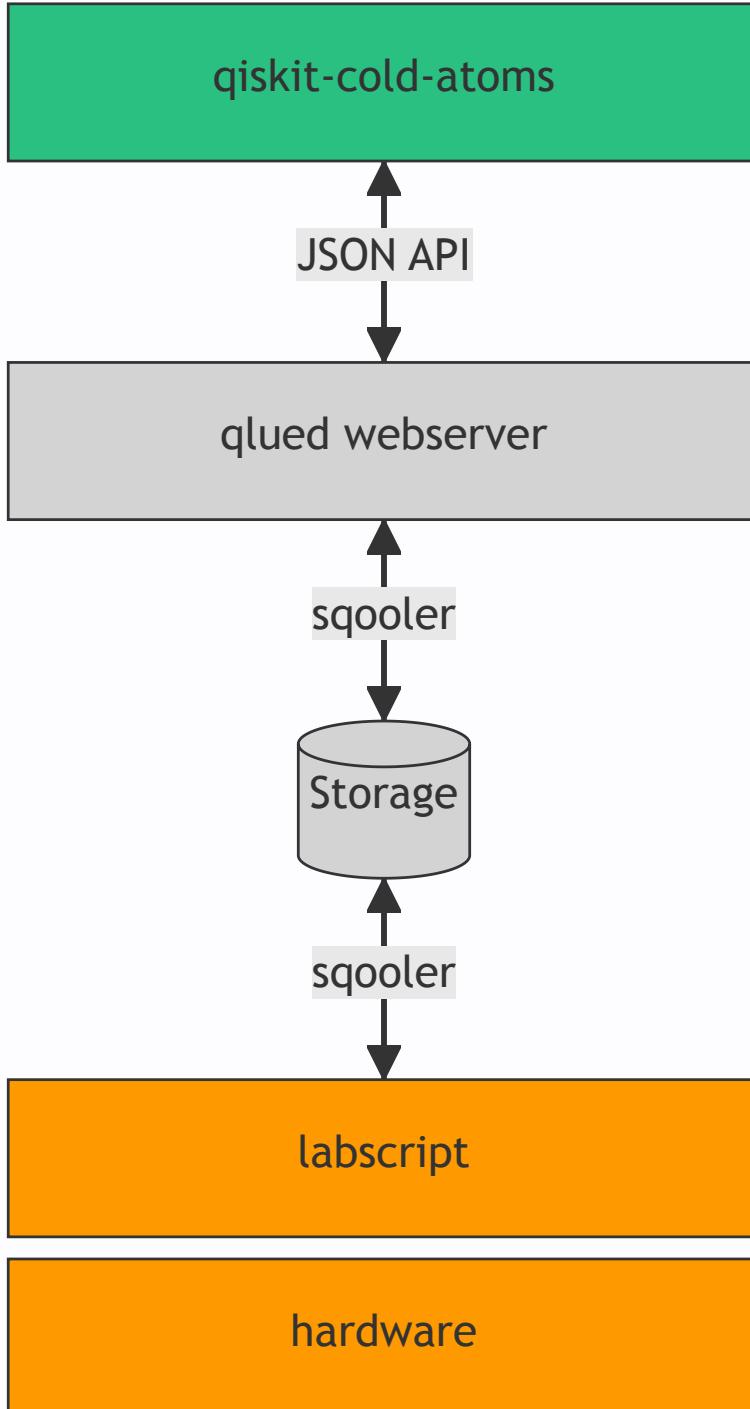
# Alice sends the json

```
{  
  "experiment_0": {  
    "instructions": [  
      ("rlx", [0], [np.pi]),  
      ("rlx", [0], [np.pi]),  
      ("measure", [1], []),  
      ("measure", [1], []),  
    ],  
    "num_wires": 2,  
    "shots": 500,  
    "wire_order": "sequential",  
  },  
}
```

- Alice sent a json file to the webserver.
- The webserver validates it and sends it to the queue of the cloud backend.
- All of this is done within the [glued](#) framework.

# Bob executes the job

- Bob pulls the job from the queue to his control PC.
- He executes the job on the cold atom device, e.g. with labscript.
- The results are sent back to the storage.



# Summary

- Alice was able to write quantum circuits in the widely used QISKit.
- Bob could control his machine through labscript software.
- Both could communicate through an open source involving qlued and sqooler .