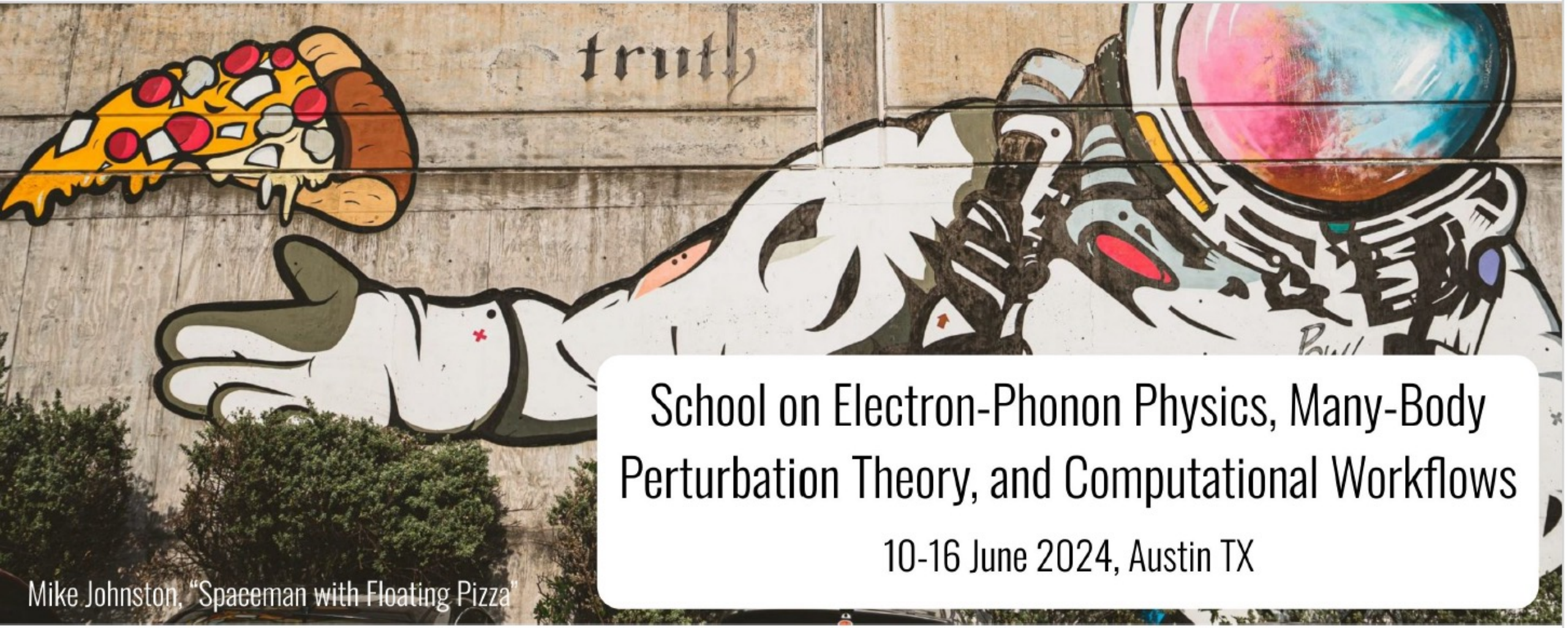


Mike Johnston, "Spaceman with Floating Pizza"

School on Electron-Phonon Physics, Many-Body Perturbation Theory, and Computational Workflows

10-16 June 2024, Austin TX

U.S. DEPARTMENT OF ENERGY

NSF

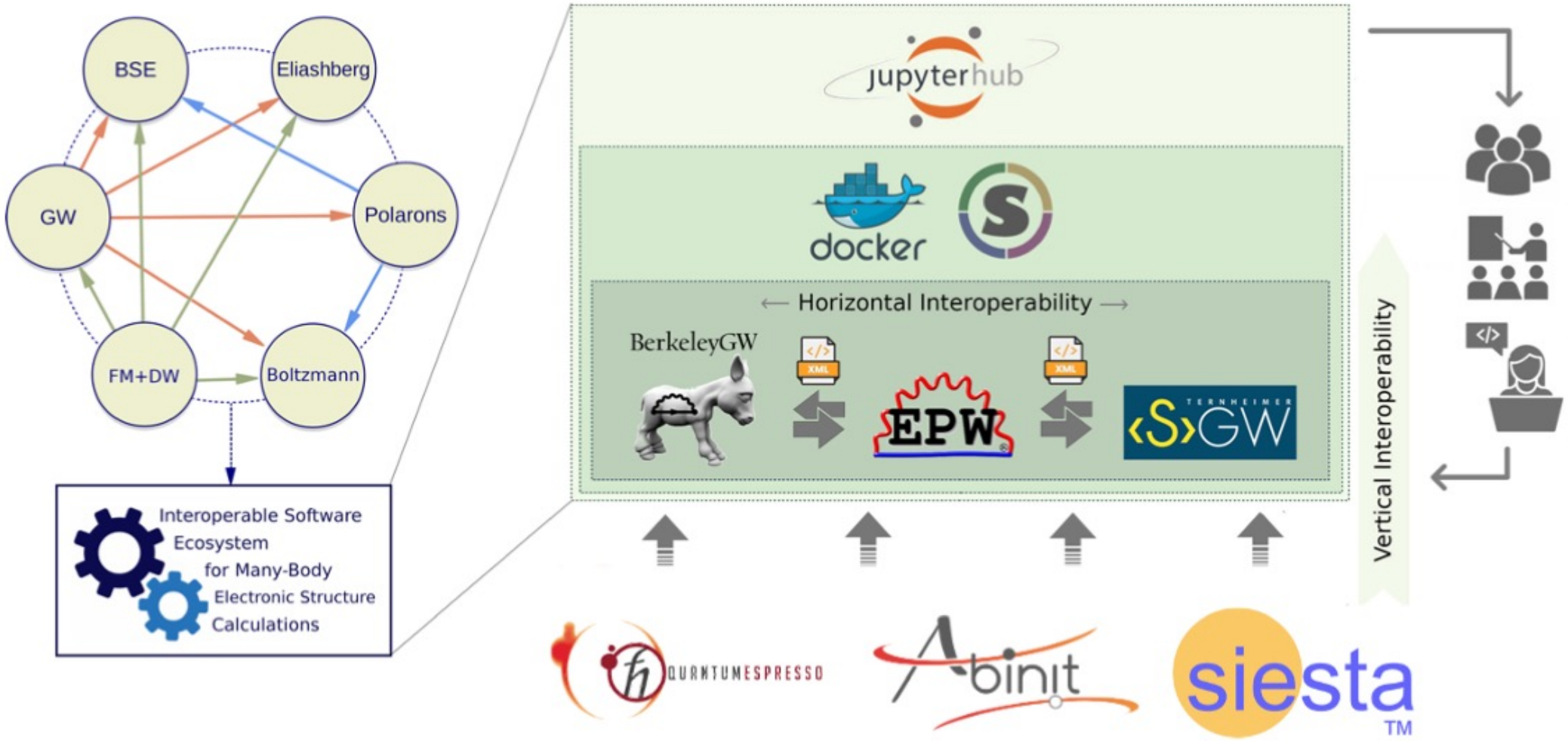
TACC
TEXAS ADVANCED COMPUTING CENTER

ODEN

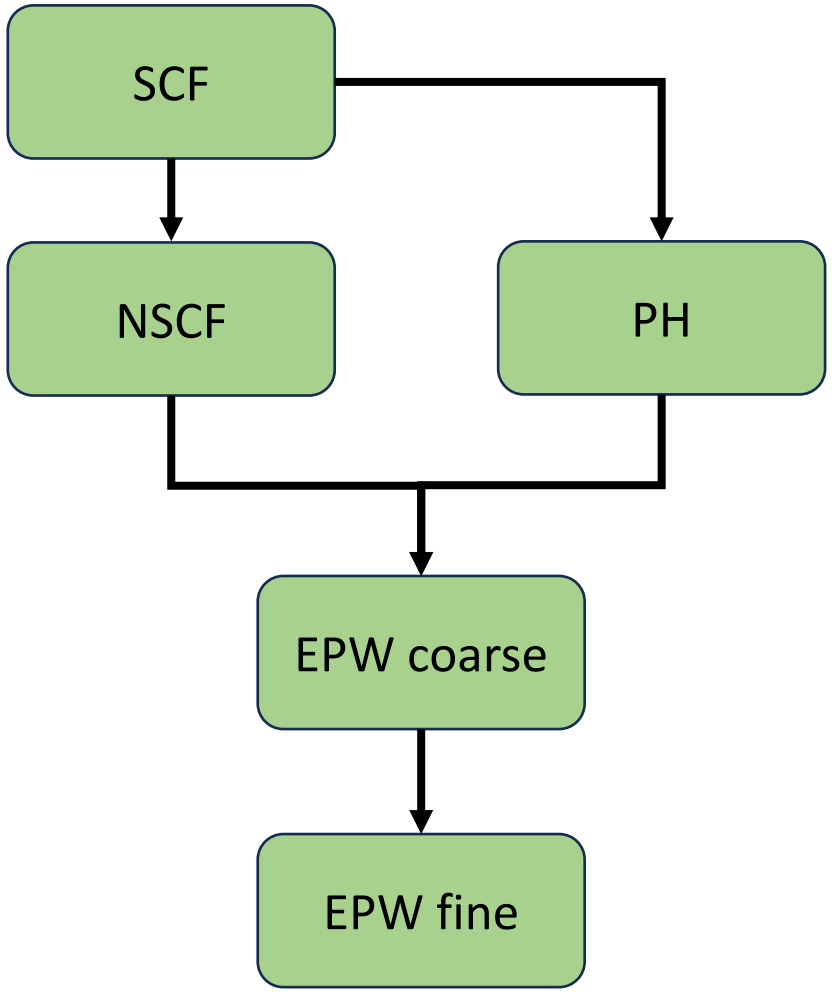
Automated workflows using EPWpy

Sun.2.Tiwari and Cucco

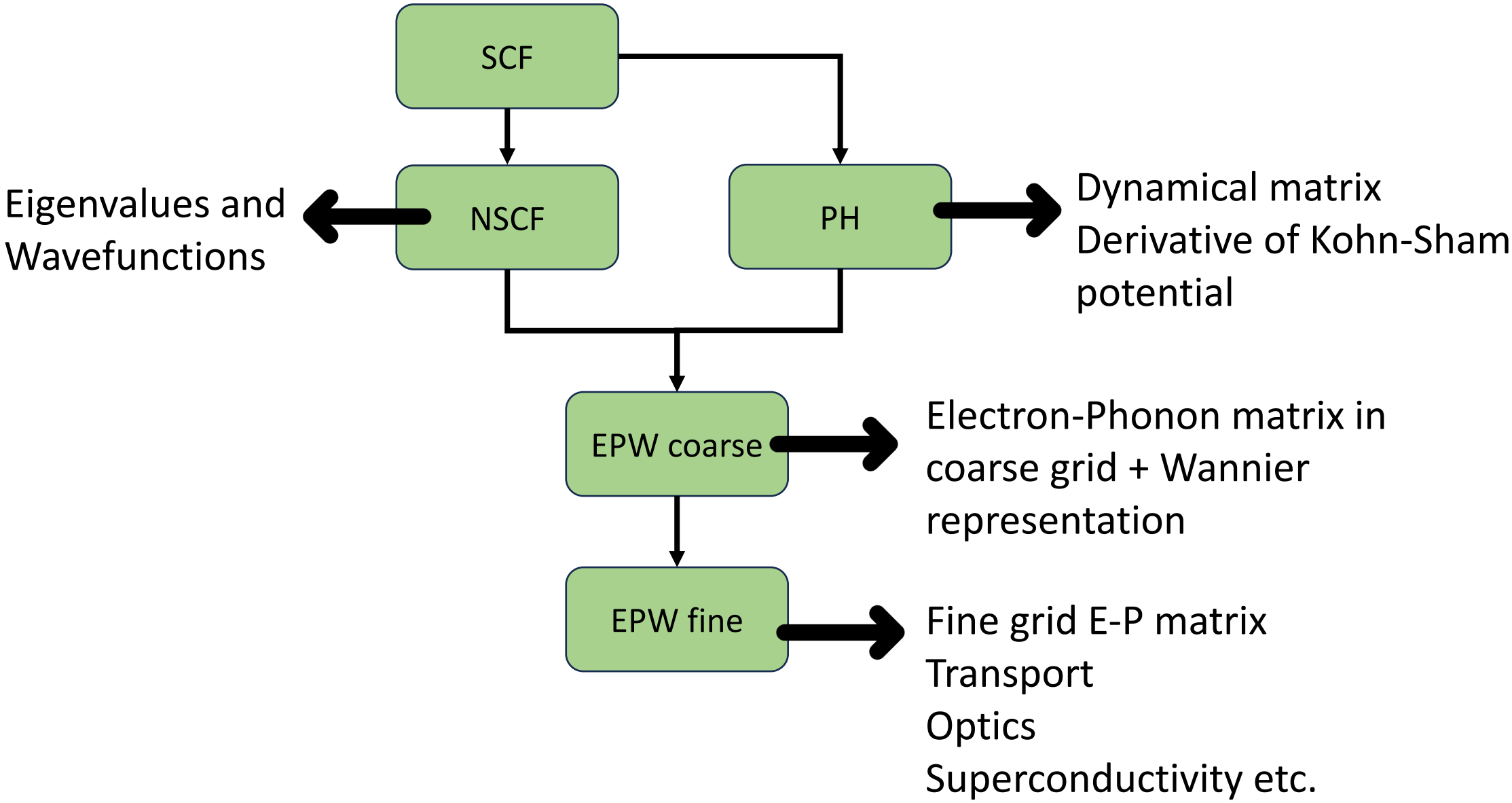
Enabling interoperable software environment



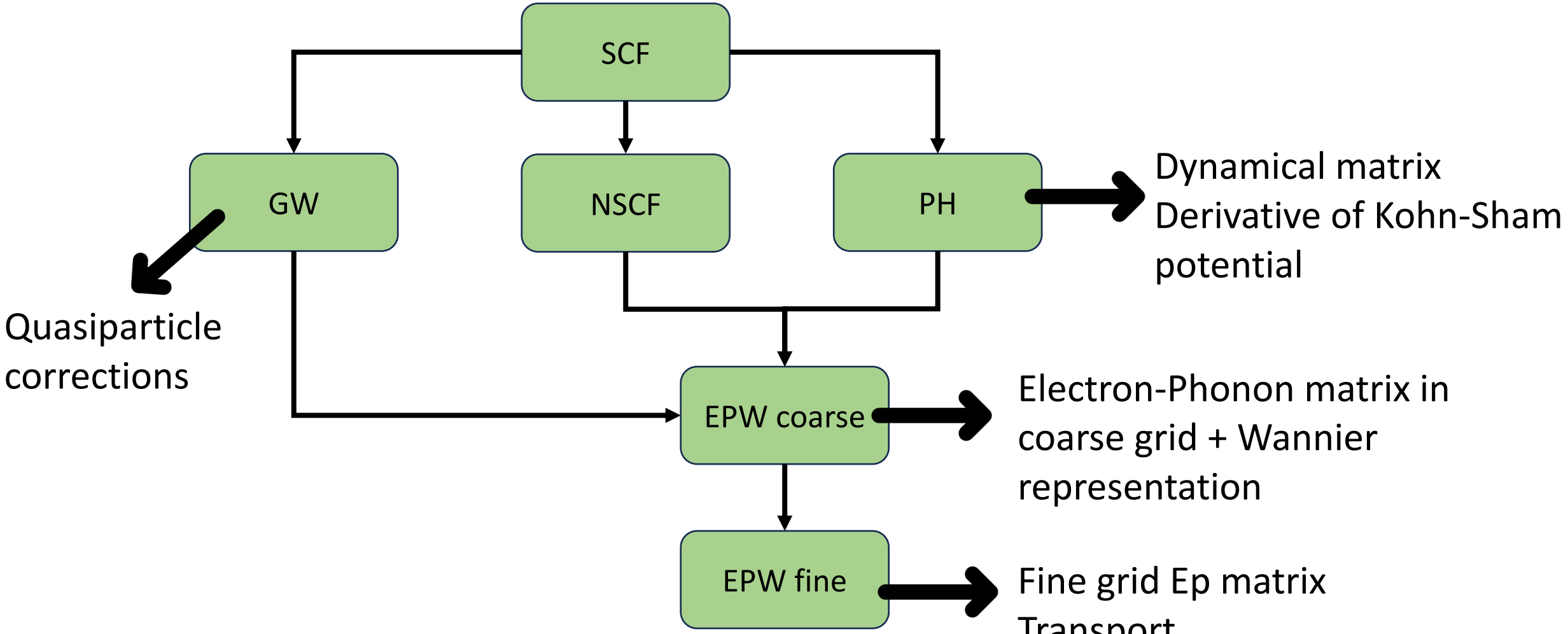
A typical workflow for EPW



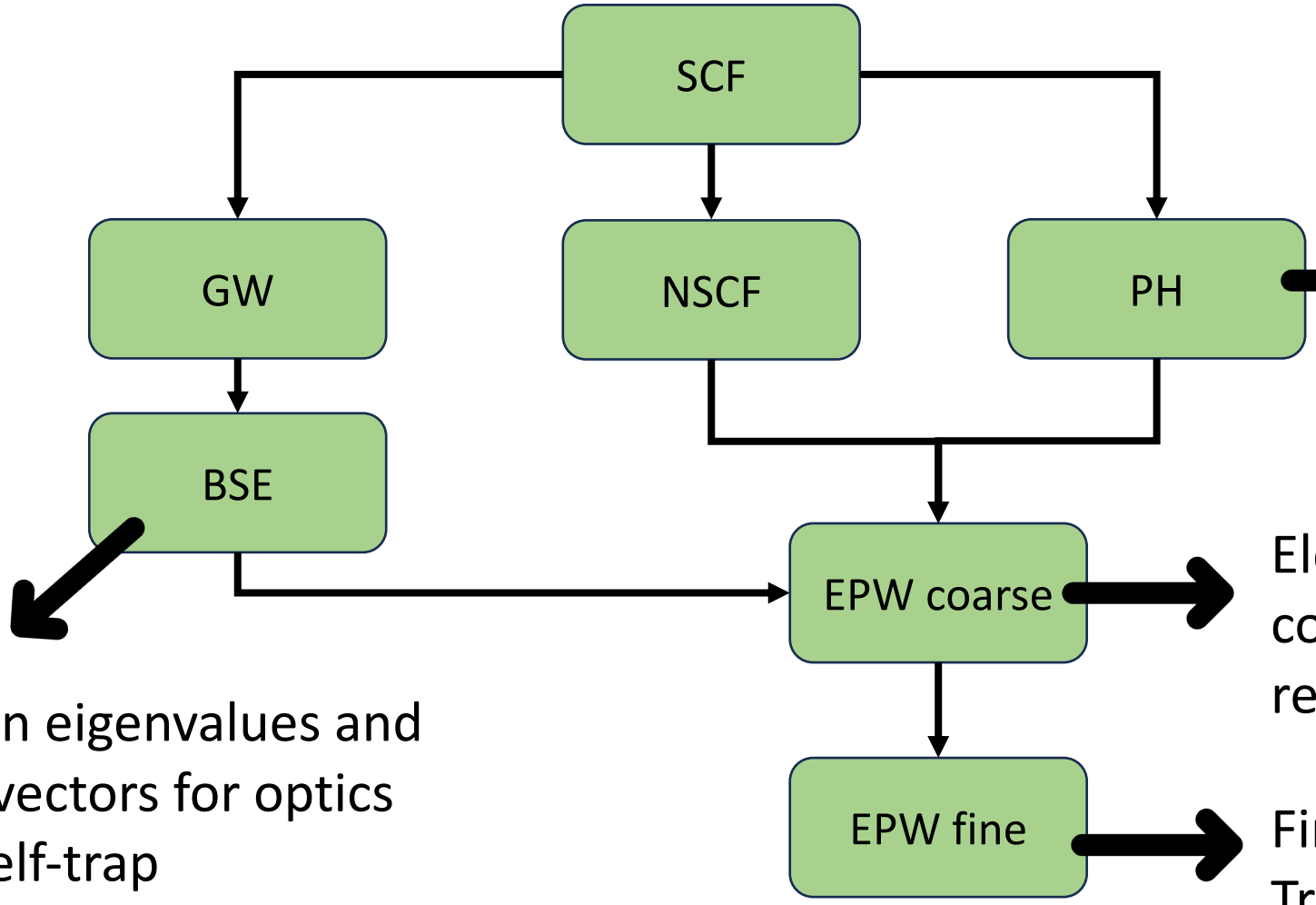
A typical workflow for EPW



A typical workflow for EPW



A typical workflow for EPW



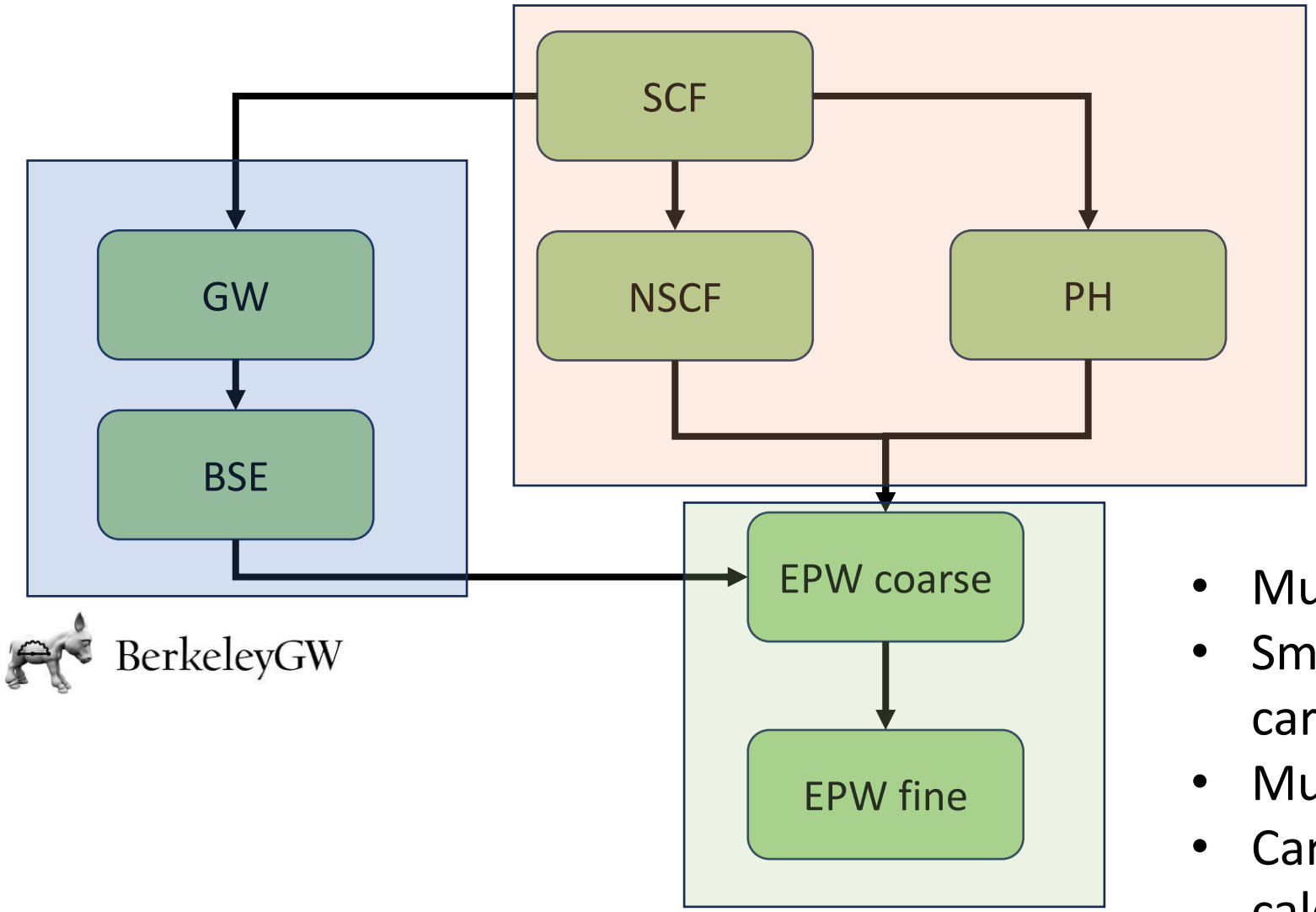
Dynamical matrix
Derivative of Kohn-Sham potential

Electron-Phonon matrix in coarse grid + Wannier representation

Exciton eigenvalues and eigenvectors for optics and self-trap

Fine grid Ep matrix
Transport
Optics
Superconductivity etc.

A typical workflow for EPW+GW+BSE



BerkeleyGW

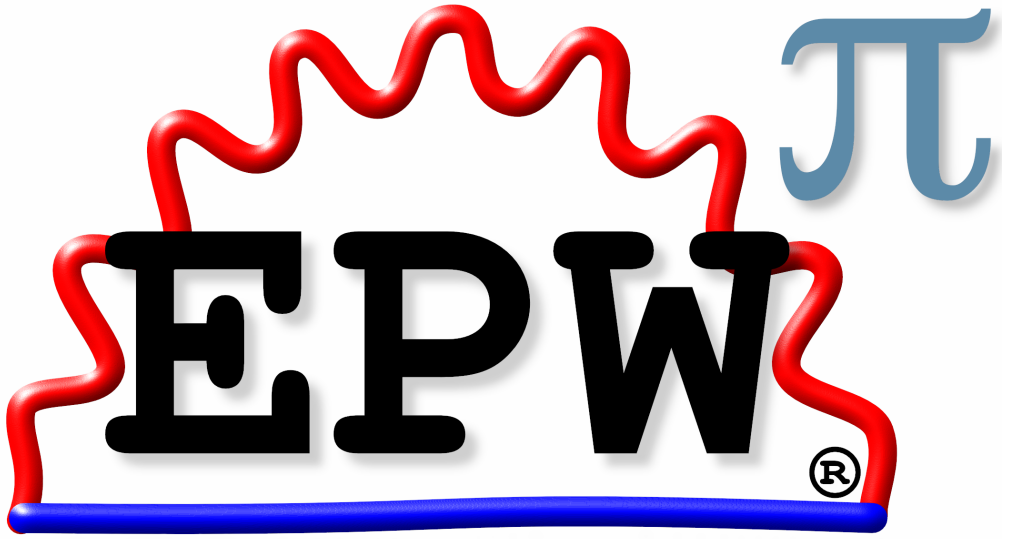
QUANTUMESPRESSO

EPW

- Multitude of calculations
- Small mistake in any step carries over
- Multiple file transfer
- Careful ordering of calculations
- Steep learning curve

EPWpy

- Python package
  - Wrap various codes to work with EPW
  - Easy to use and develop
  - Streamline calculations
    - Parallelly run multiple calculations
  - High level access to data (for analysis)
  - Makes it possible to use EPW using a Jupyter notebook



A typical EPWpy workflow

```
silicon=EPWpy({'prefix':prefix,
               'restart_mode':'\'from_scratch\'',
               'ibrav':2,
               'nat':2,
               'calculation':'\'scf\'',
               'atomic_species':['Si'],
               'mass':[28.0855],
               'atoms':['Si','Si'],
               'ntyp':1,
               'pseudo':['Si.upf'],
               'ecutwfc':'40',
               'ecutrho':'160',
               'celldm(1)':'10.262',
               'verbosity':'high',
               'pseudo_dir':'\''+str(pseudo)+'\''
                },
               code=QE,
               env='ibrun')
```

→ Material definition

```
silicon.scf(control={'calculation':'\'bands\''},
            system={'nbnd':12},
            electrons={'conv_thr':'1E-11'},
            kpoints={'kpoints':[['0.5', '0.5', '0.5', '20'],
                                ['0.0','0.0','0.0','20'],
                                ['0.5000','0.2500','0.7500','20']],
                     'kpoints_type':'{crystal_b}'},
            name='bs')
#####################################

silicon.prepare(20,type_run='bs')
silicon.run(4,type_run='bs')
```

→ Preparation and calculation

- Definition not needed, most of the things are automated

Three-step process

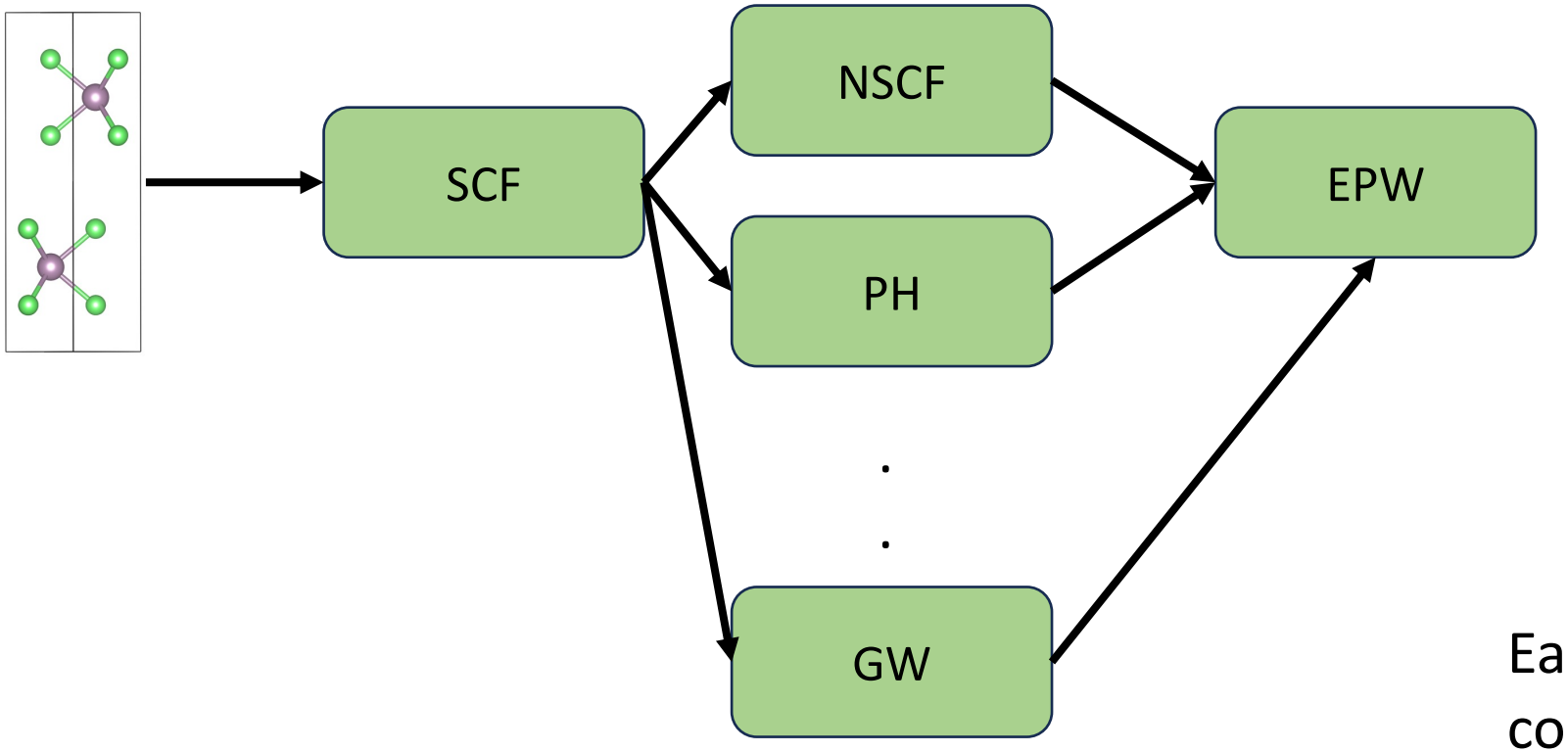
```
silicon.scf(control={'calculation':'\'bands\''},
            system={'nbnd':12},
            electrons={'conv_thr':'1E-11'},
            kpoints={'kpoints':[['0.5', '0.5', '0.5', '20'],
                                ['0.0','0.0','0.0','20'],
                                ['0.5000','0.2500','0.7500','20']],
                     'kpoints_type':'{crystal_b}'},
            name='bs')
#######################################
silicon.prepare(20,type_run='bs')
silicon.run(4,type_run='bs')
```

1. Calling the method prepares and writes inputs

2. Prepares files needed for calculation

3. Runs the calculation

Adding blocks of calculation

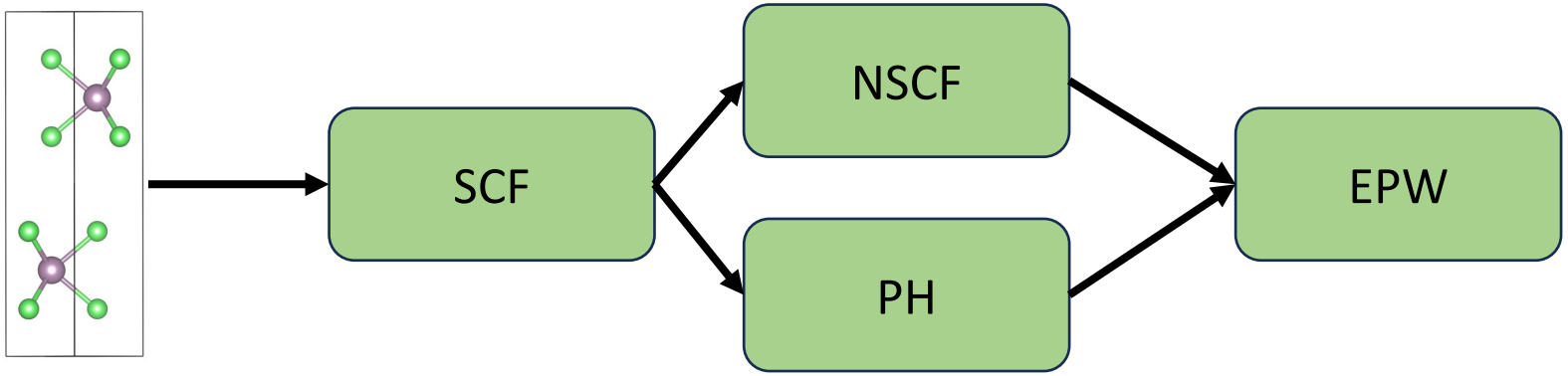


Easily added as a block or combination of blocks

In [ ]:

```
#Prepare and GW calculation
silicon.GW(GW={'nbnd':20})
silicon.run(16,'GW')
#Prepare and run epsilon
silicon.epsilon(epsilon={'restart':' ', 'degeneracy_check_override':' '})
silicon.run(16,'epsilon')
#Prepare and run Sigma calculation
silicon.sigma(sigma={'band_index_min': 4,'band_index_max':12})
silicon.run(16,'sigma')
```

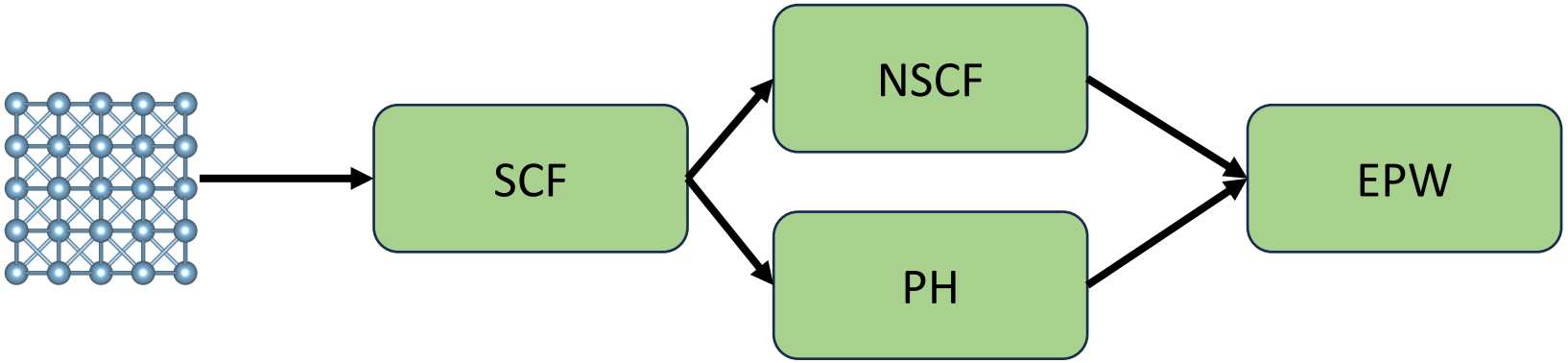
Multiple materials workflow



ZG notebook

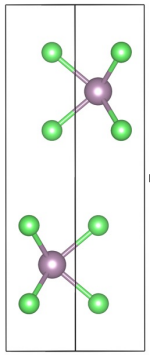
.
.
.

Many materials in workflow parallelly run



Order of running
Not important

Folder structure



Material

Outer folders

SCF

NSCF

PH

EPW

Inner folders

All transfers are handled by EPWpy

Analysis of results

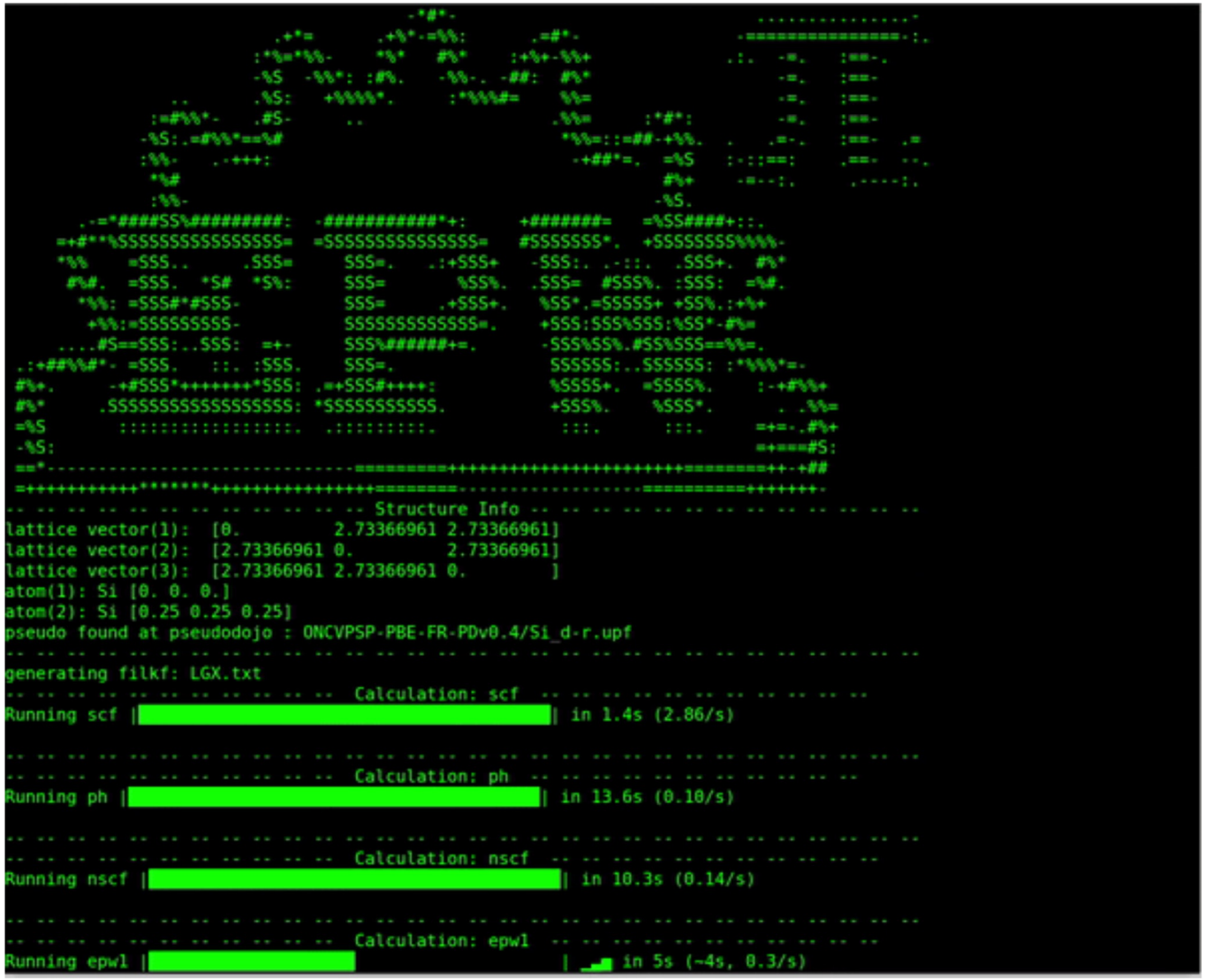
```
############Kramers-Kronig-Analysis ###################################


silicon.type_run='qdabs'
silicon.default_epw_input['meshnum']='7'

silicon.eps0=10.0
silicon.temp=300.0
eps1=silicon.eps1
eps2=silicon.eps2
omega=silicon.omega
nr=silicon.nr
plt.plot(omega,eps1)
plt.plot(omega,eps2)
```

- Provides data as variables of the material class
- Further calculations using notebook

EPWpy in action



Some friends

- Reset button: <object>.reset()
- utilities package inside EPWpy: Learn in respective notebooks
- py_run and py_prepare class
  - These are independent classes and can help prototype a EPWpy run block on the fly!
- Me, Bruno and all instructors!

Today’s exercise

- We have four notebooks on ZG, transport, quasi-degenerate absorption and superconductivity
- Choose one and enjoy