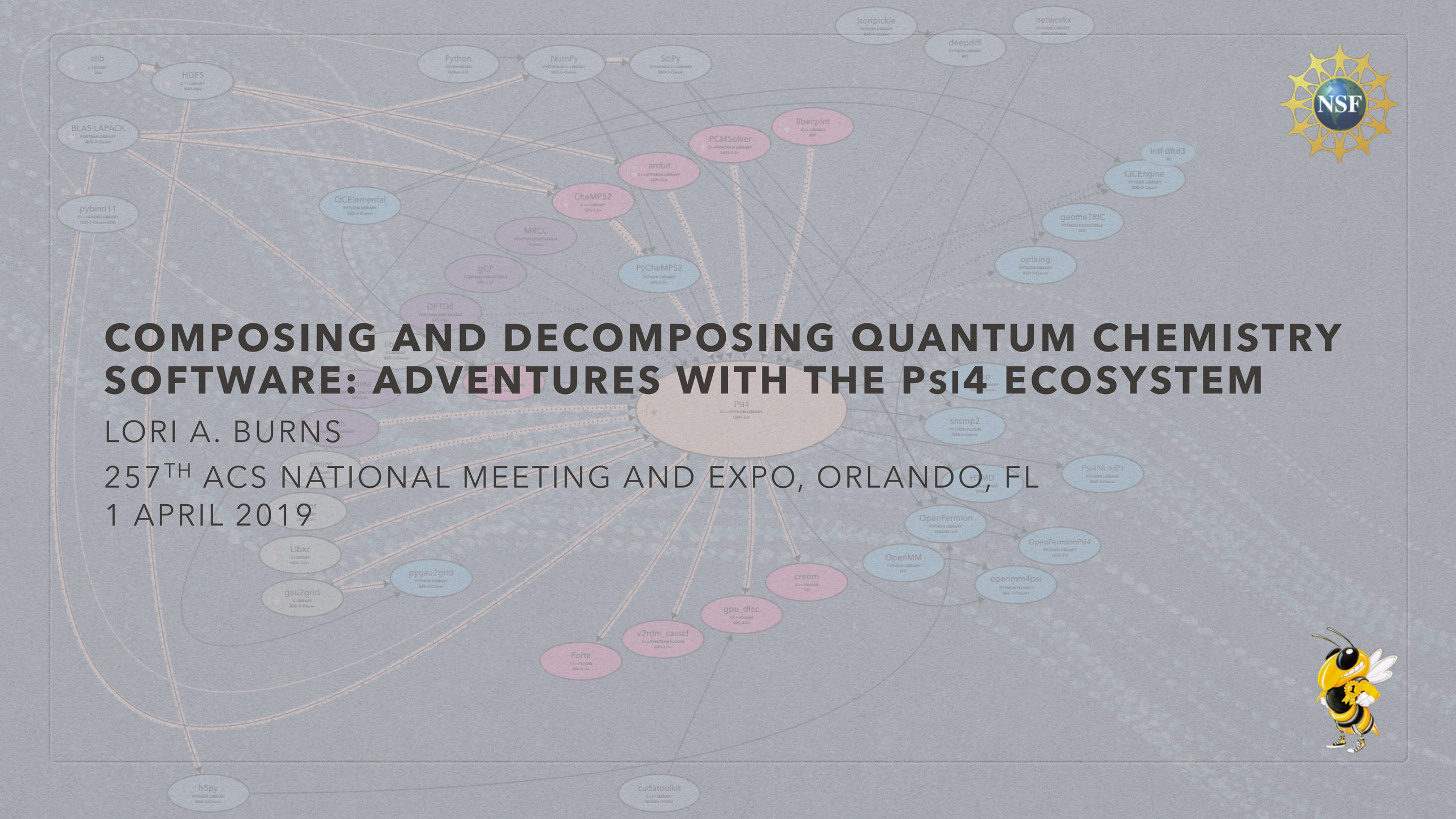


COMPOSING AND DECOMPOSING QUANTUM CHEMISTRY SOFTWARE: ADVENTURES WITH THE PSI4 ECOSYSTEM

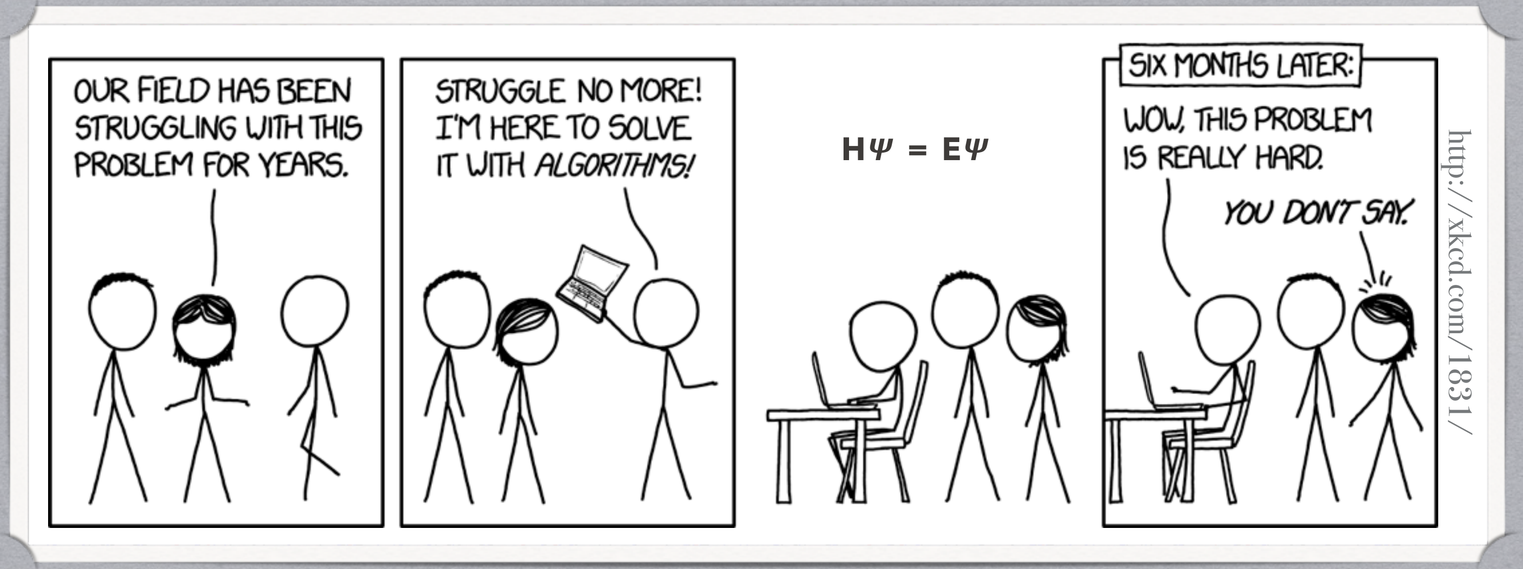
LORI A. BURNS

257TH ACS NATIONAL MEETING AND EXPO, ORLANDO, FL

1 APRIL 2019

QUANTUM CHEMISTRY HAS PROBLEMS ENOUGH

LET OTHERS SOLVE THE REST



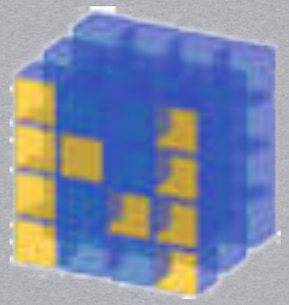
QUANTUM CHEMISTRY HAS PROBLEMS ENOUGH

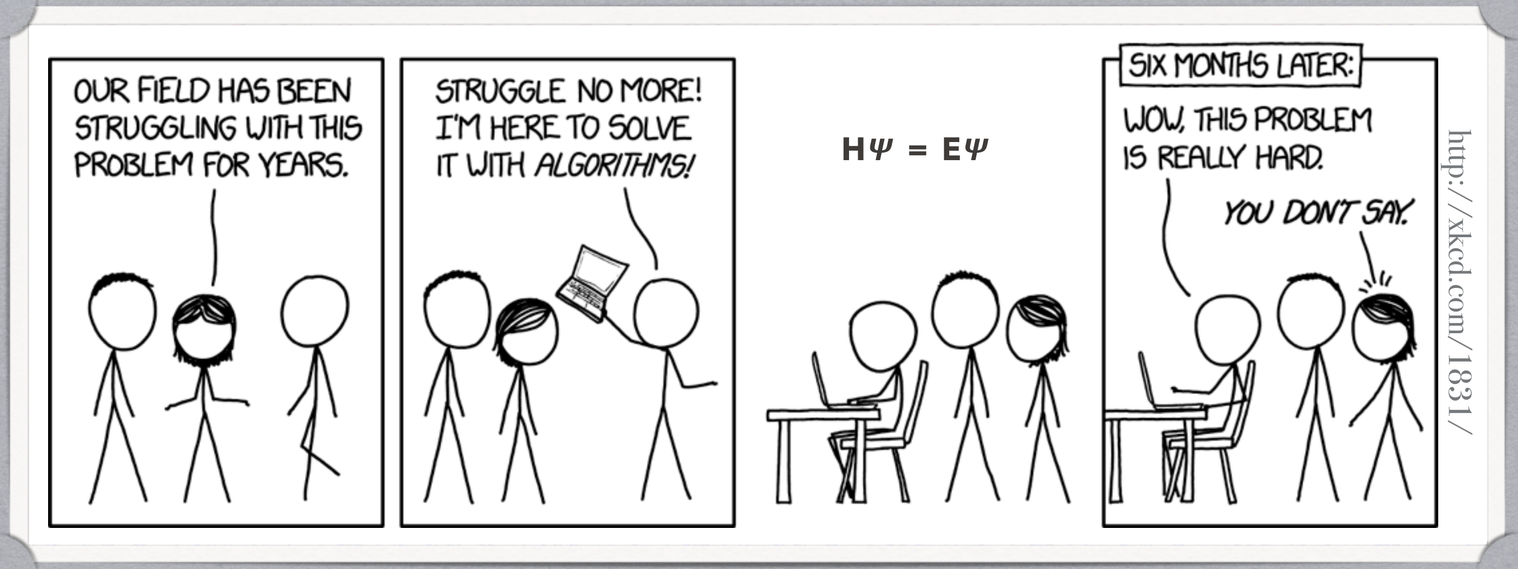
LET OTHERS SOLVE THE REST

python

CMake

pybind11





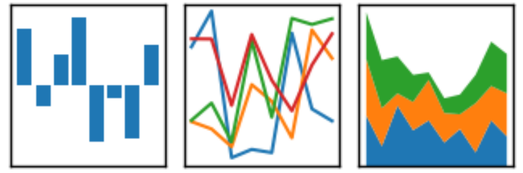
HDF

L A P A C K
L -A P -A C -K
L A P A -C -K
L -A P -A C K
L A -P -A C K
L -A -P A C -K

git

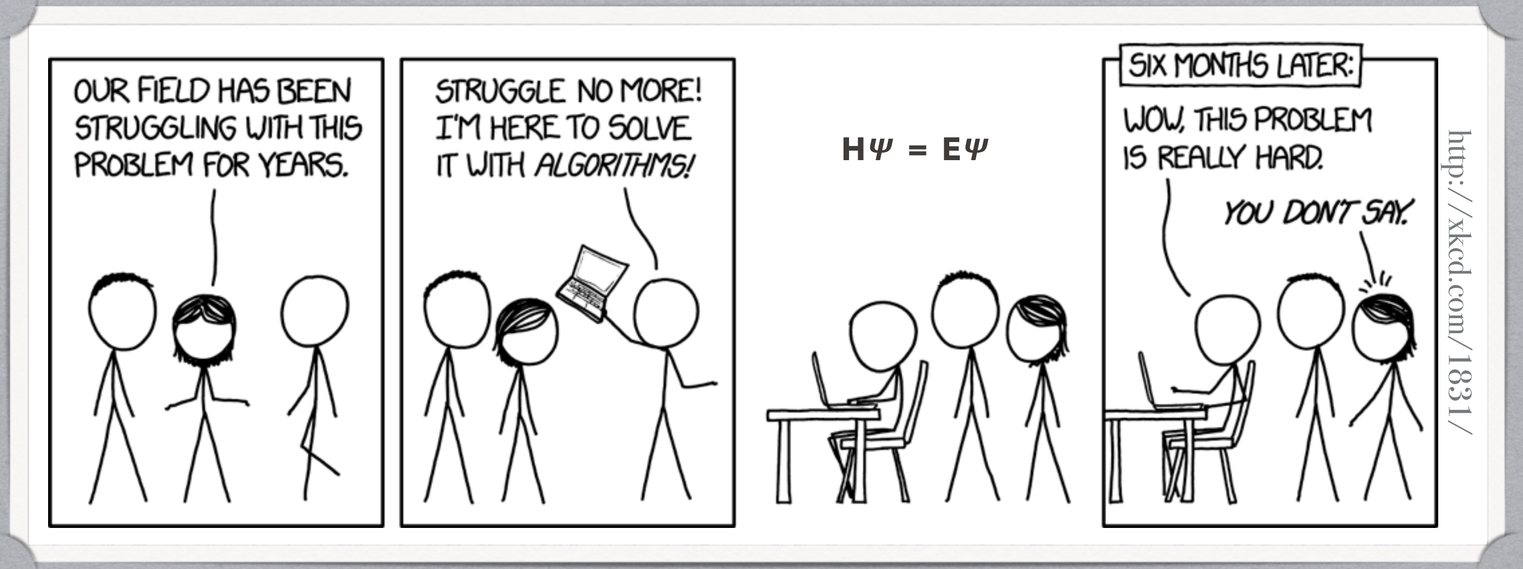
pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$



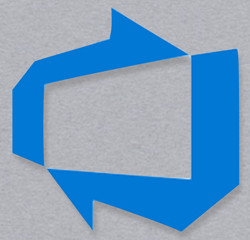
QUANTUM CHEMISTRY HAS PROBLEMS ENOUGH

LET OTHERS SOLVE THE WORKFLOW

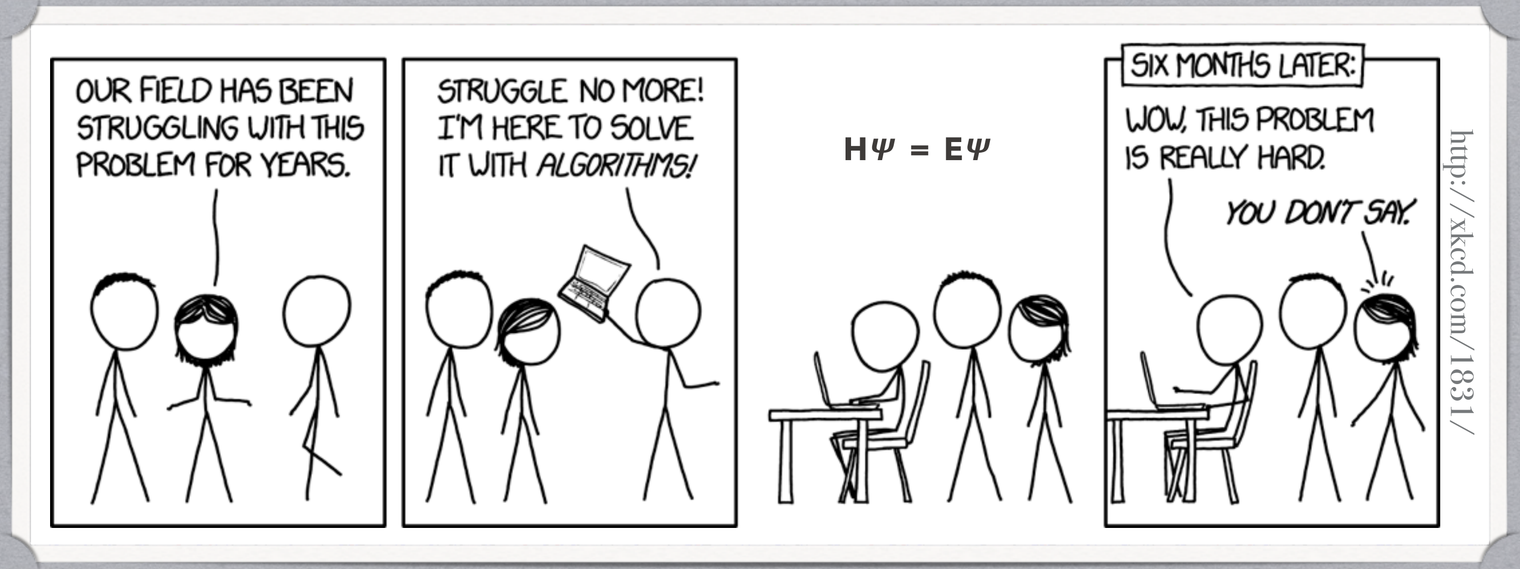


QUANTUM CHEMISTRY HAS PROBLEMS ENOUGH

LET OTHERS SOLVE THE WORKFLOW



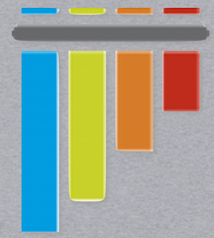
ANACONDA CLOUD



jupyter

CONDA-FORGE

DASK



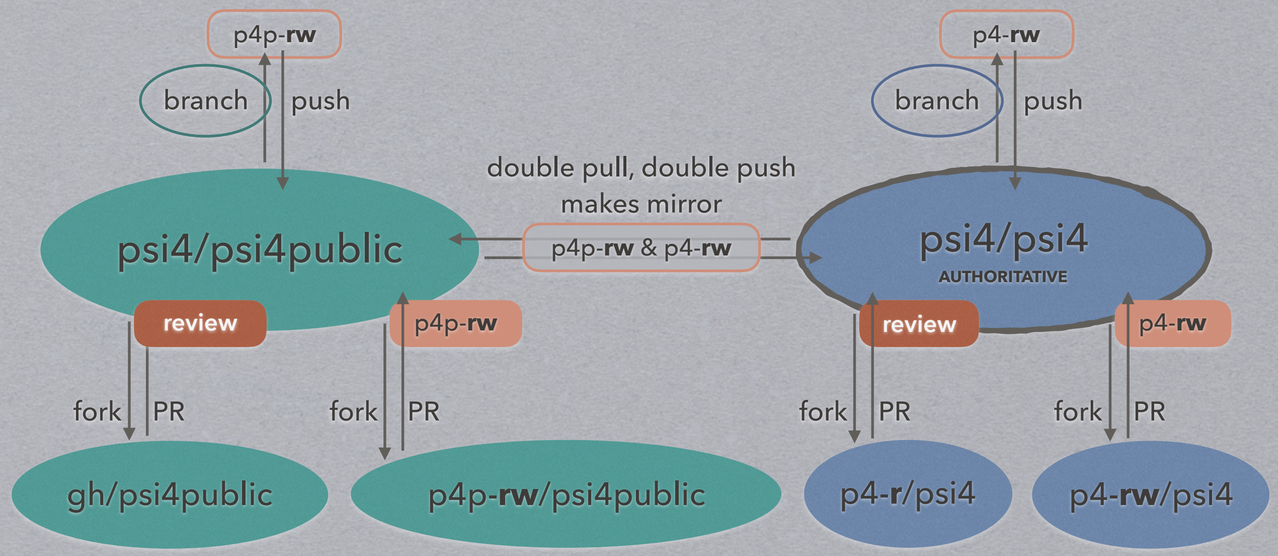
GIT WORKFLOW

UNGUIDED TO GUIDED

GIT WORKFLOW

UNGUIDED TO GUIDED

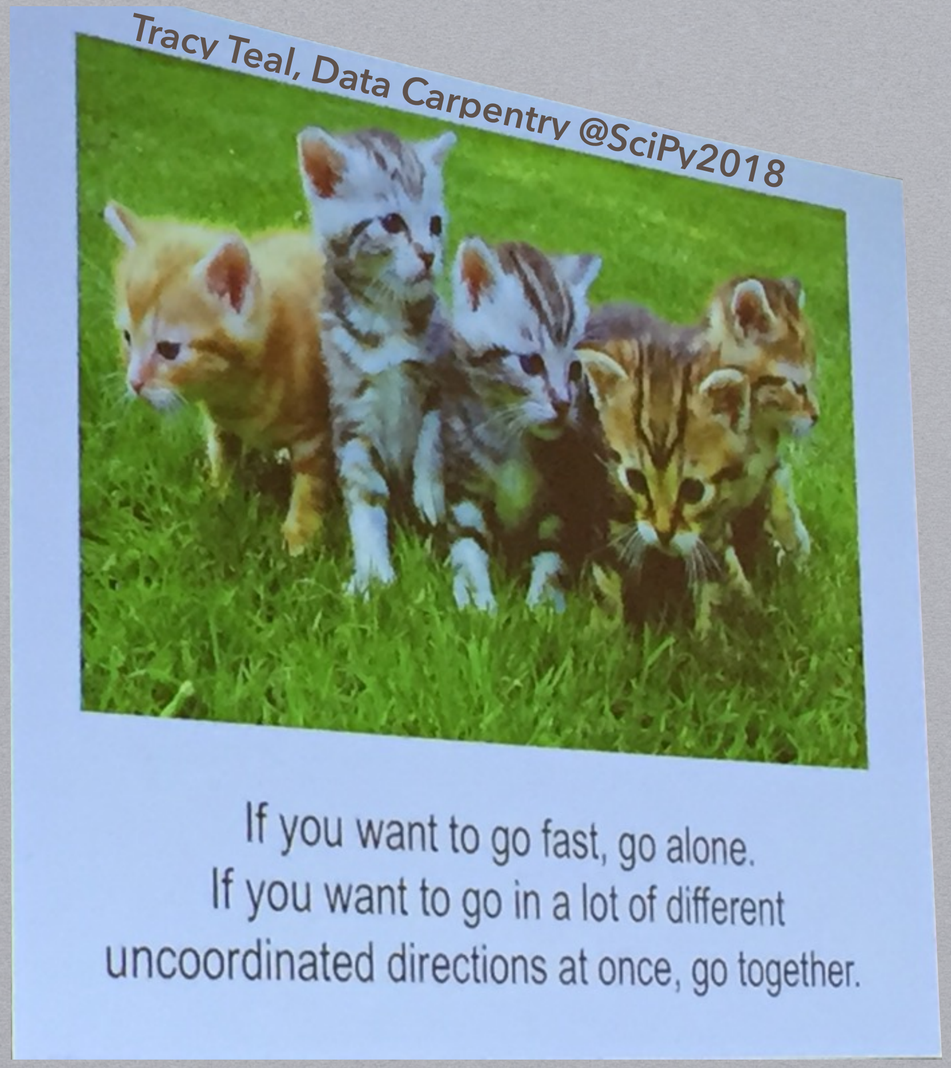
2015



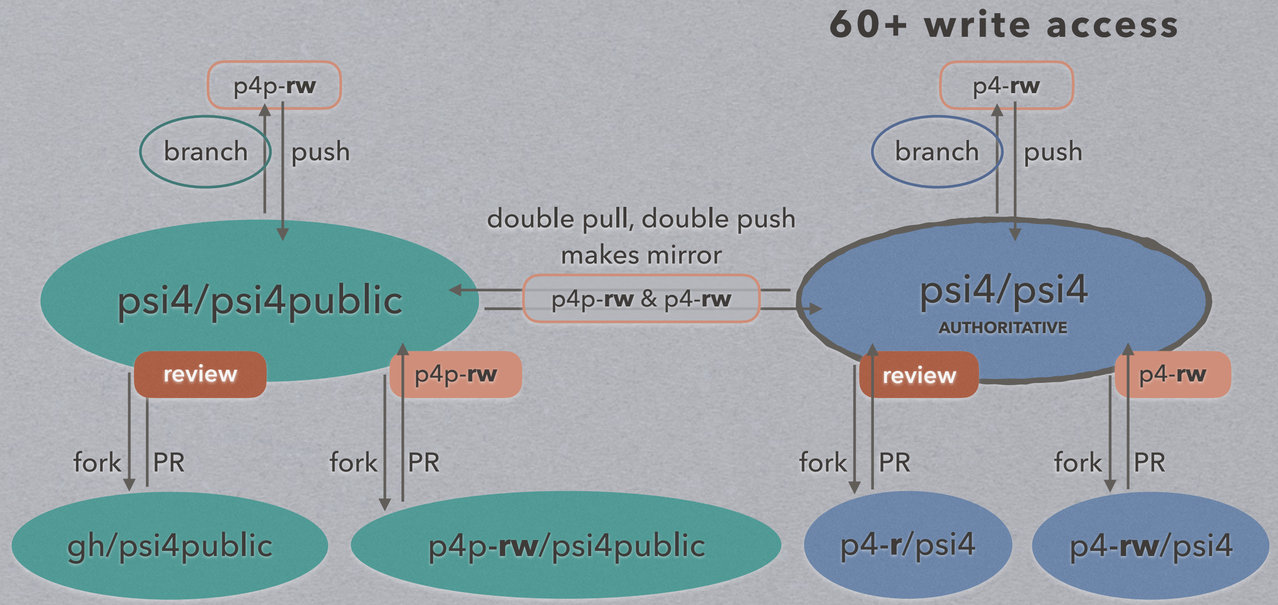
public

private

GIT WORKFLOW
UNGUIDED TO GUIDED



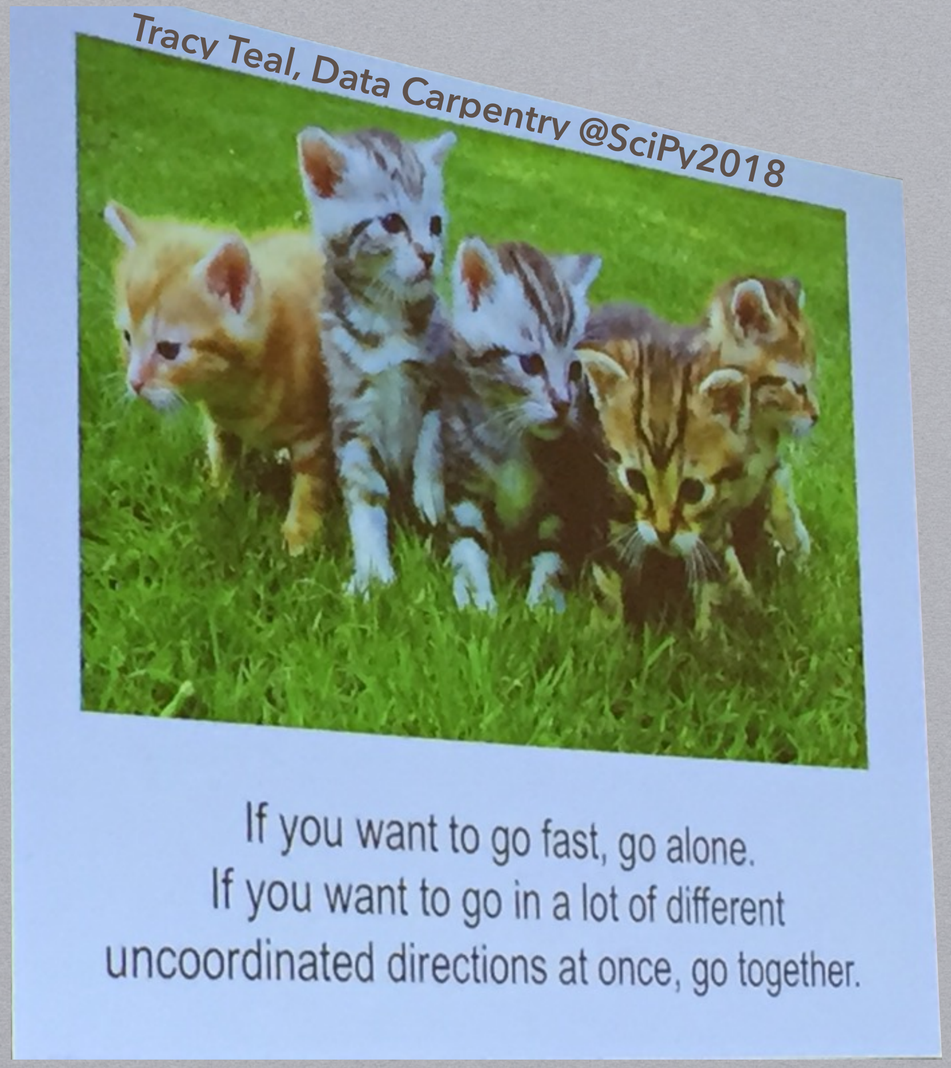
2015



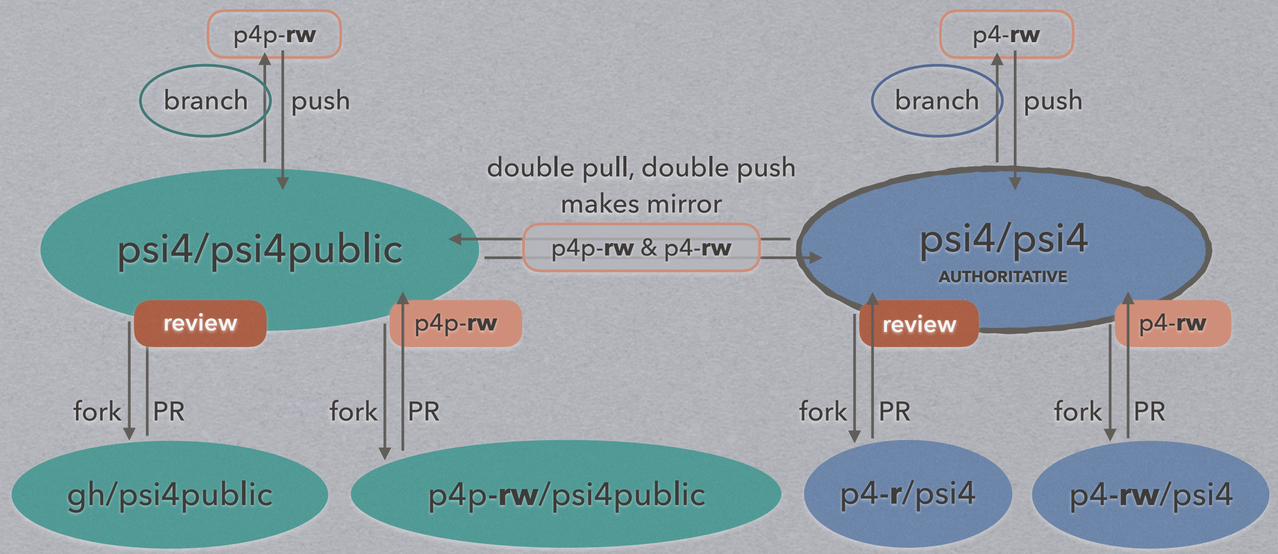
public

private

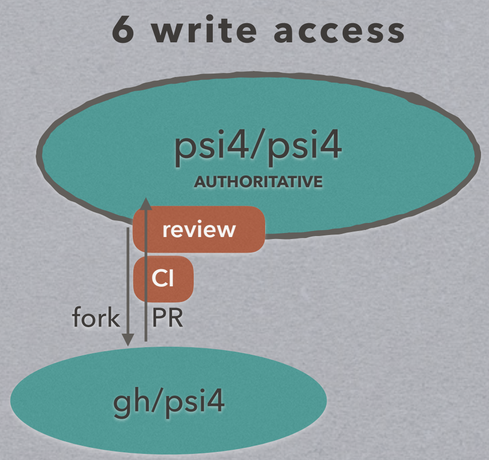
GIT WORKFLOW
UNGUIDED TO GUIDED



2015



2019



- **PUBLIC**, not private development.
- **FORKING WORKFLOW**, readily transferable to other open-source projects. Prefer rebase over merge.
- **PR EARLY**, so feedback can be given.
- **SAFETY NET** new contributors needn't worry about harming the project.

public *private*

CODE REVIEW

MAINTAINERS ARE KEYSTONE SPECIES

- **THREE REVIEWS** (not including the proposer) must approve before PR merge, and *every* change occurs through PR.
- **DELOCALIZATION** initiated to ensure (by bot) that devs at different institutions actively approve direction of project.
- **COMPREHENSIVE** appraisal since each reviewer has strengths and specialities.
- **ECOSYSTEM** protects the interests of developing downstream research projects
- **INSTITUTIONAL KNOWLEDGE** keeps core developers familiar with whole project.

NOT A QUORUM FOR CHANGING PSI

"There's a perverse effect where, **the more successful you are, the more you get "punished" with GitHub notifications.**"

@nolanlawson,
"What it Feels Like to be an Open Source Maintainer"

https://snarky.ca/setting-expectations-for-open-source-participation/

BUS FACTOR

RESILIANCY OF PROJECT

- **NON-CORE** devs contributed 166/335 PRs for v1.3.
- **CONTRIBUTORS** nearly doubled since private → public

PLoP 1994 https://github.com/Robsteranium/bus-factor http://gittrends.io/#/

BUS FACTOR

RESILIANCY OF PROJECT

**NUMBER OF DEVELOPERS WHO,
IF SUDDENLY ABSENT, STALL A PROJECT**

- **NON-CORE** devs contributed 166/335 PRs for v1.3.
- **CONTRIBUTORS** nearly doubled since private → public

PLoP 1994 https://github.com/Robsteranium/bus-factor http://gittrends.io/#/

BUS FACTOR

RESILIANCY OF PROJECT

**NUMBER OF DEVELOPERS WHO,
IF SUDDENLY ABSENT, STALL A PROJECT**

- **NON-CORE** devs contributed 166/335 PRs for v1.3.
- **CONTRIBUTORS** nearly doubled since private → public

Project	Bus Factor
Linux	164
Clang	9
NumPy	4
Psi4	4
TensorFlow	2
jQuery	2
OpenSSL	1

PLoP 1994 https://github.com/Robsteranium/bus-factor http://gittrends.io/#/

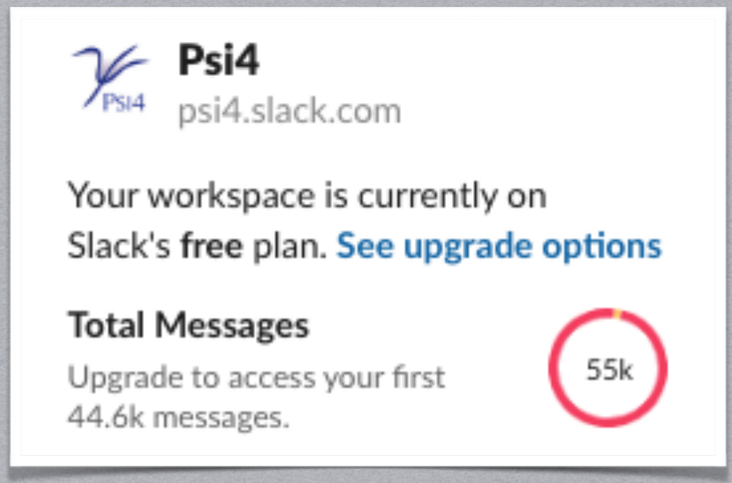
COMMUNICATION

GITHUB AND SLACK

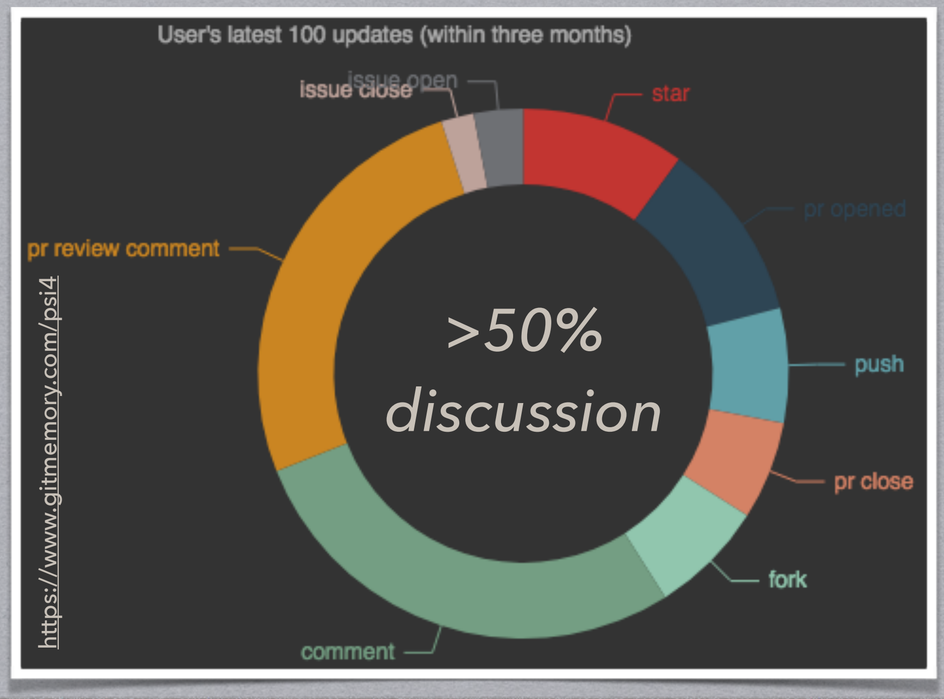
- **LOWER** communication barriers.
  - **OPEN** invitation off GitHub
  - **NEW-DEV** needing guidance
  - **ORIG-DEV** for historical context
- **GAUGE** consensus quickly w/ polls, emoji.
- **EU/US** usually always a core dev awake.
- **CORE-DEV** chat has been continuous for 2.5 years through three programs.

COMMUNICATION

GITHUB AND SLACK



- **LOWER** communication barriers.
  - **OPEN** invitation off GitHub
  - **NEW-DEV** needing guidance
  - **ORIG-DEV** for historical context
- **GAUGE** consensus quickly w/ polls, emoji.
- **EU/US** usually always a core dev awake.
- **CORE-DEV** chat has been continuous for 2.5 years through three programs.

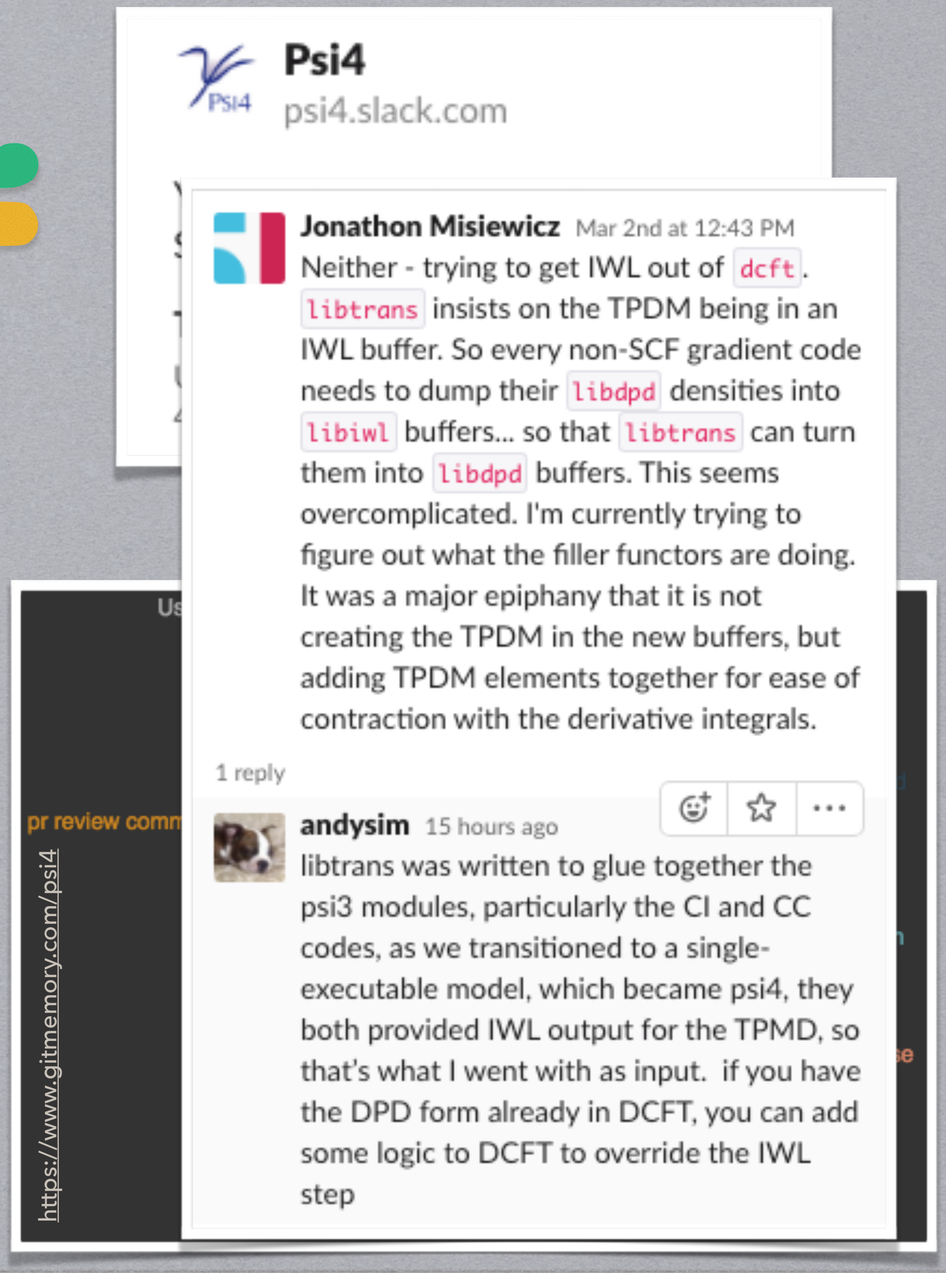


COMMUNICATION

GITHUB AND SLACK

- **LOWER** communication barriers.
  - **OPEN** invitation off GitHub
  - **NEW-DEV** needing guidance
  - **ORIG-DEV** for historical context
- **GAUGE** consensus quickly w/ polls, emoji.
- **EU/US** usually always a core dev awake.
- **CORE-DEV** chat has been continuous for 2.5 years through three programs.

dev chat on slack

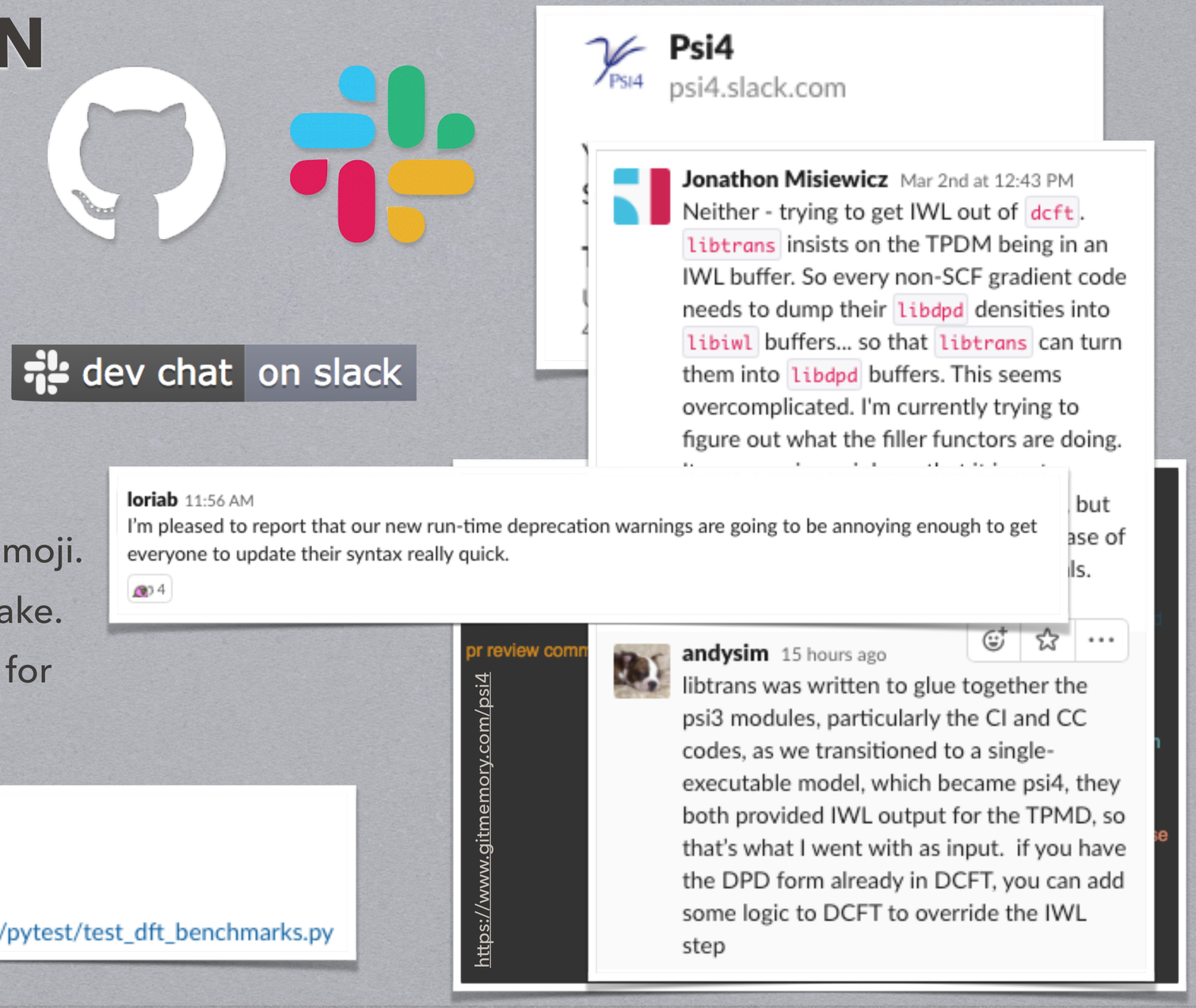


COMMUNICATION

GITHUB AND SLACK

- **LOWER** communication barriers.
  - **OPEN** invitation off GitHub
  - **NEW-DEV** needing guidance
  - **ORIG-DEV** for historical context
- **GAUGE** consensus quickly w/ polls, emoji.
- **EU/US** usually always a core dev awake.
- **CORE-DEV** chat has been continuous for 2.5 years through three programs.

dev chat on slack



CONTINUOUS INTEGRATION

TRAVIS-CI & AZURE

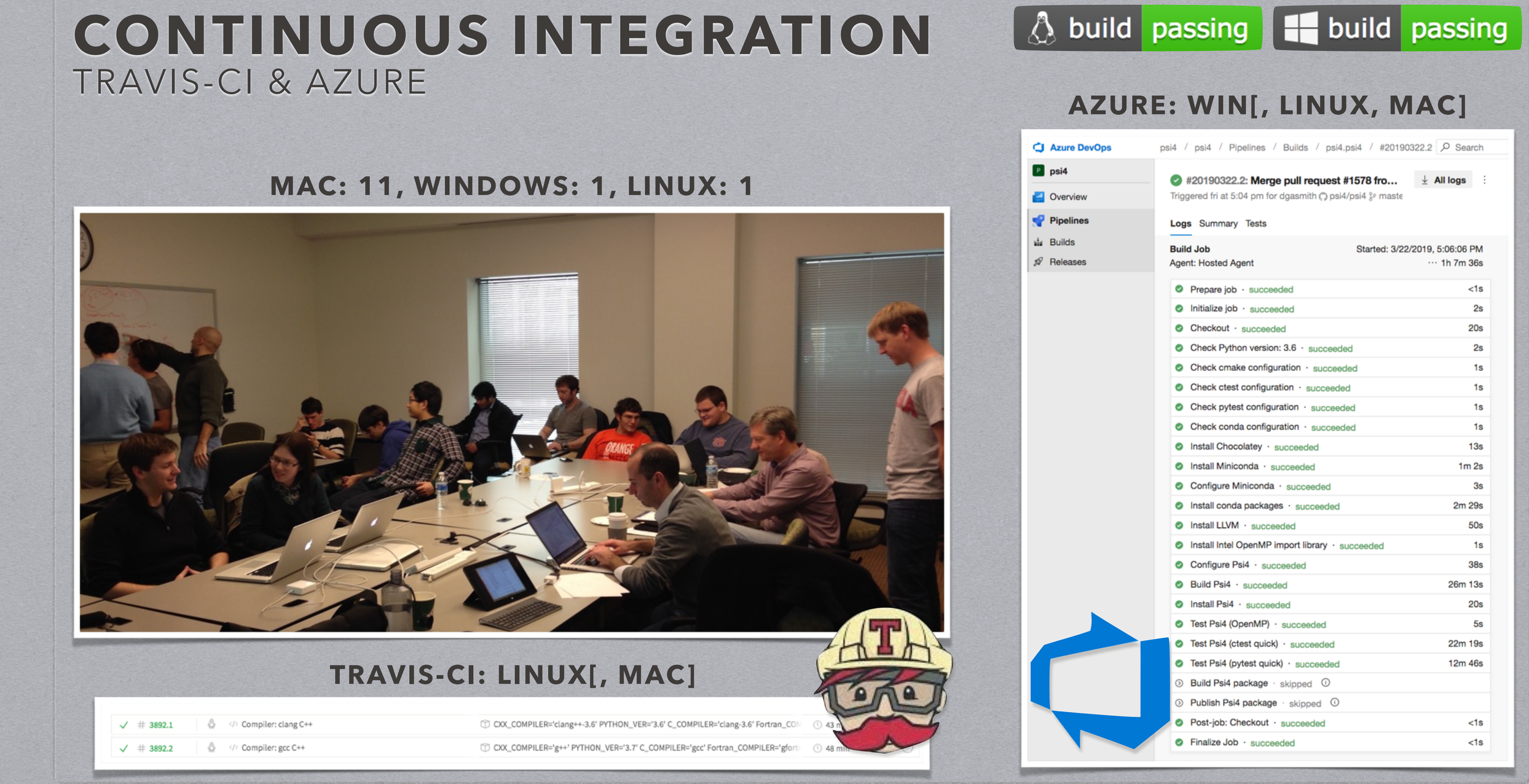
CONTINUOUS INTEGRATION

TRAVIS-CI & AZURE

MAC: 11, WINDOWS: 1, LINUX: 1

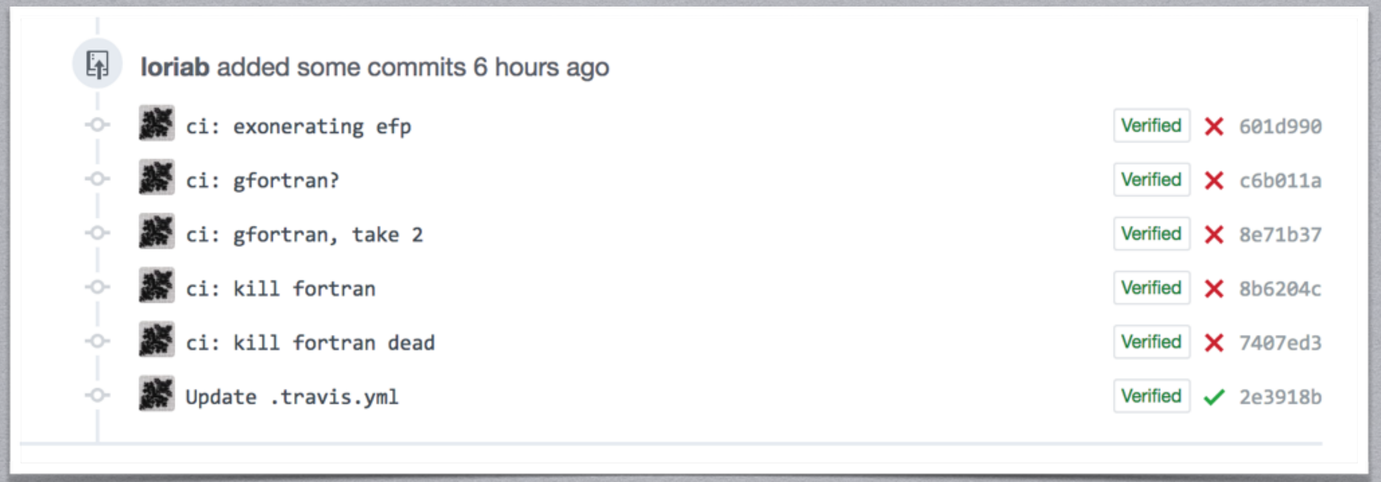
CONTINUOUS INTEGRATION

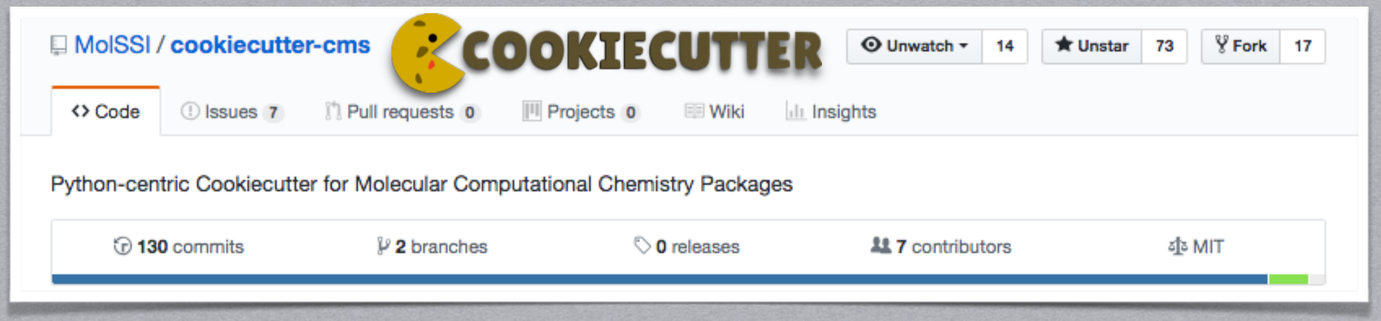
TRAVIS-CI & AZURE



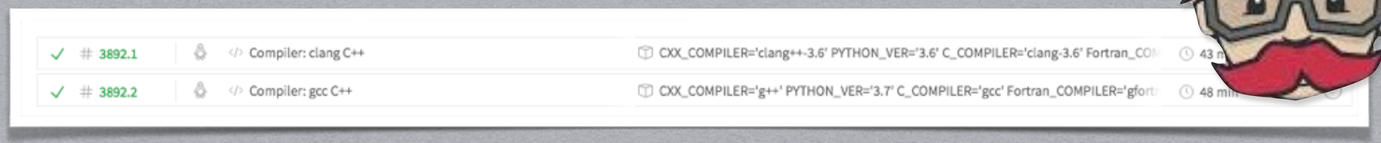
CONTINUOUS INTEGRATION

TRAVIS-CI & AZURE



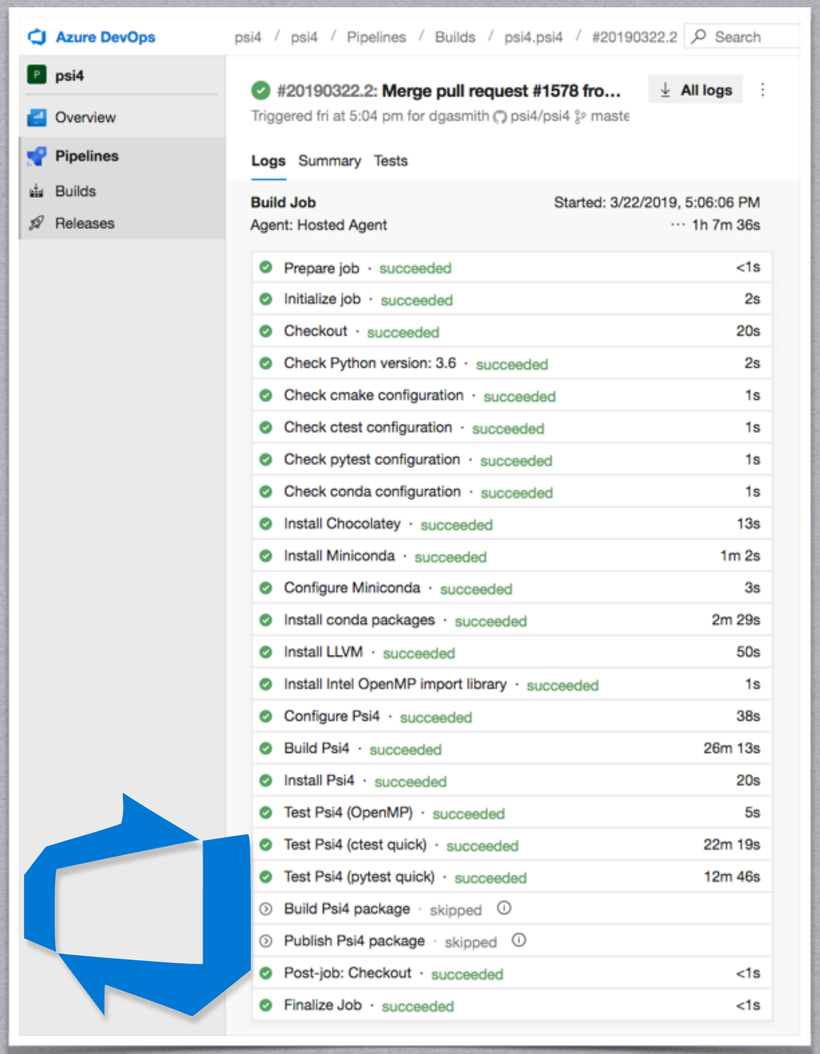


TRAVIS-CI: LINUX[, MAC]



build passing build passing

AZURE: WIN[, LINUX, MAC]



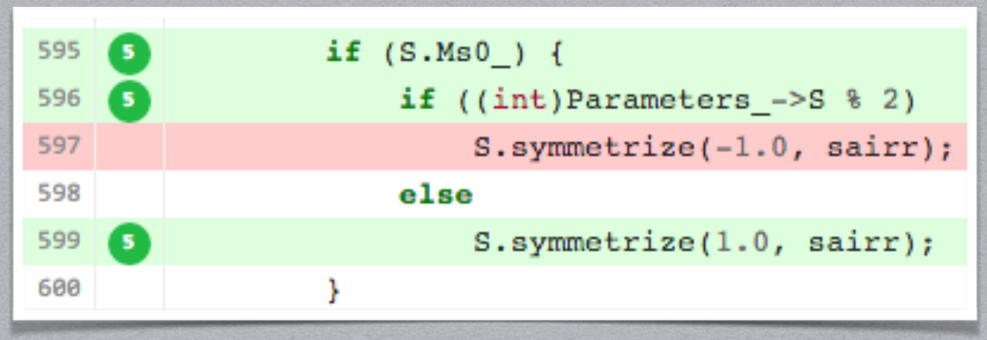
DYNAMIC ANALYSIS

CODECOV

- **BUILDS** and runs test suite with different tools for different languages. Upload line tabulations to codecov site for analysis.
- **ALERTS** on lines never hit by test suite.
- **ERROR TESTING** is common area to fall short.
- **GITHUB** integration as PR check available.
- **LOCATED** bugs by mending skipped lines.

DYNAMIC ANALYSIS

CODECOV

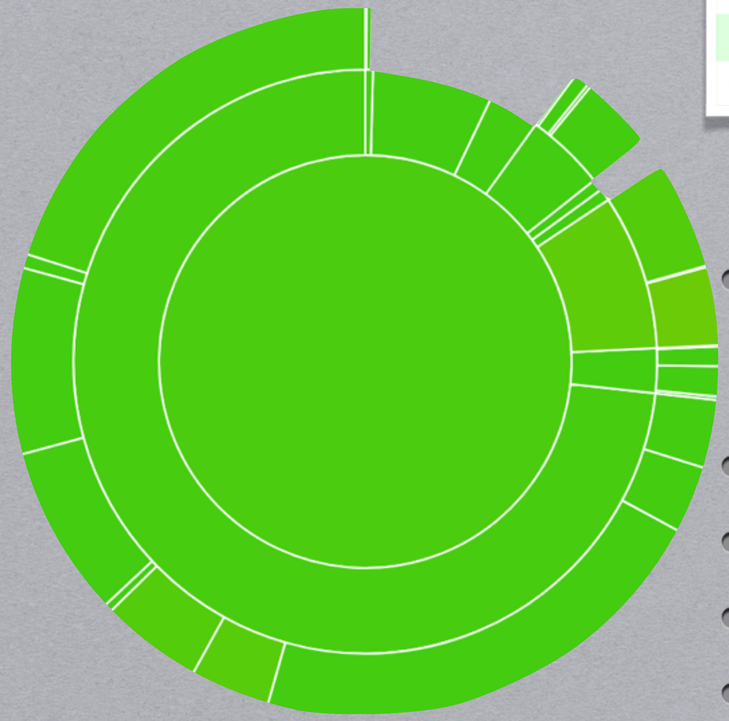


- **BUILDS** and runs test suite with different tools for different languages. Upload line tabulations to codecov site for analysis.
- **ALERTS** on lines never hit by test suite.
- **ERROR TESTING** is common area to fall short.
- **GITHUB** integration as PR check available.
- **LOCATED** bugs by mending skipped lines.

DYNAMIC ANALYSIS

CODECOV

QCElemental, 99.6%



```
595  5          if (S.Ms0_) {
596  5              if ((int)Parameters_->S % 2)
597                     S.symmetrize(-1.0, sairr);
598                 else
599  5                 S.symmetrize(1.0, sairr);
600             }
```

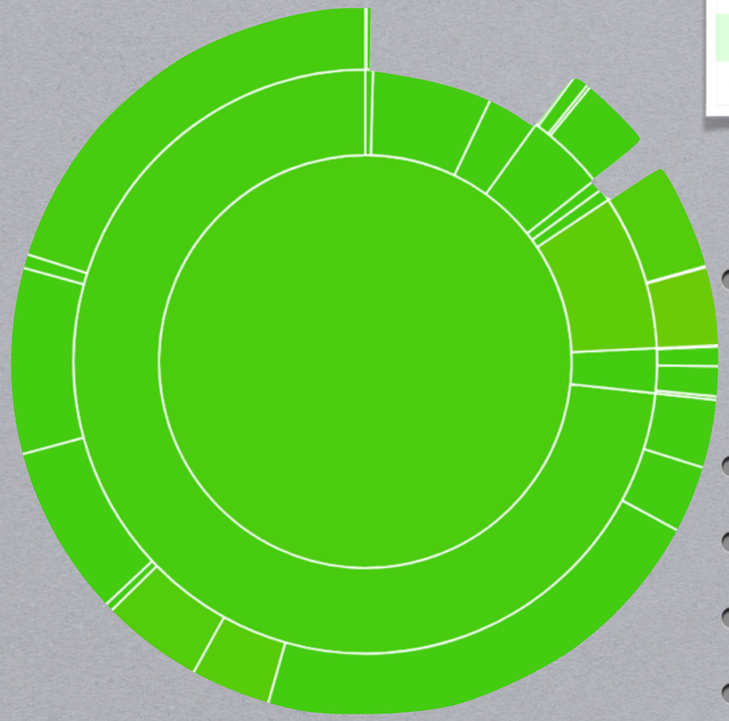
- **BUILDS** and runs test suite with different tools for different languages. Upload line tabulations to codecov site for analysis.
- **ALERTS** on lines never hit by test suite.
- **ERROR TESTING** is common area to fall short.
- **GITHUB** integration as PR check available.
- **LOCATED** bugs by mending skipped lines.

DYNAMIC ANALYSIS

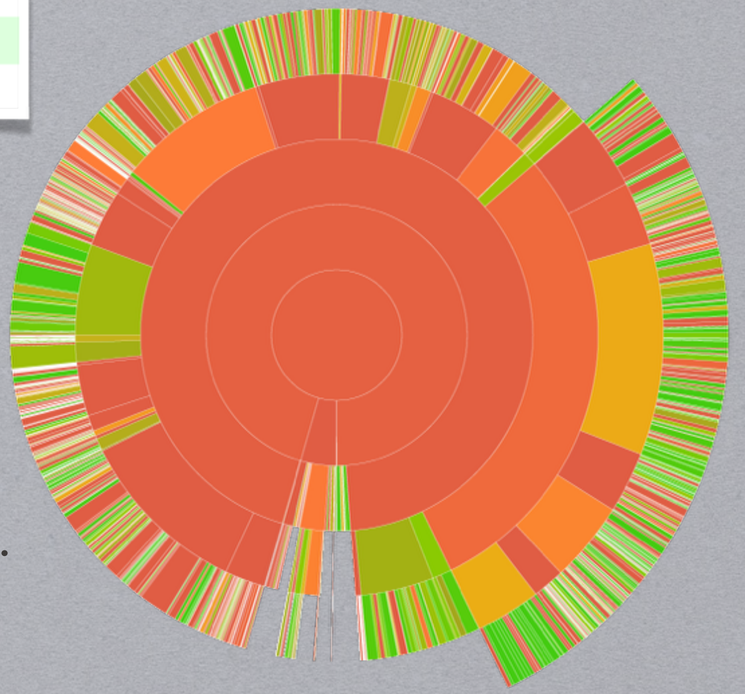
CODECOV

```
595  5      if (S.Ms0_) {
596  5          if ((int)Parameters_->S % 2)
597                 S.symmetrize(-1.0, sairr);
598             else
599  5              S.symmetrize(1.0, sairr);
600         }
```

QCElemental, 99.6%

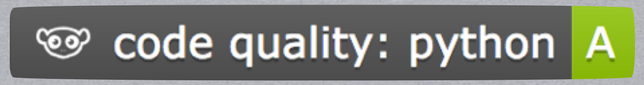


Psi4, 71.0%



- **BUILDS** and runs test suite with different tools for different languages. Upload line tabulations to codecov site for analysis.
- **ALERTS** on lines never hit by test suite.
- **ERROR TESTING** is common area to fall short.
- **GITHUB** integration as PR check available.
- **LOCATED** bugs by mending skipped lines.

STATIC ANALYSIS
LOOKS GOOD TO ME



- **BUILDS** – but does not run – repository for several languages. Runs known fault patterns on code or define your own.
- **ALERTS** ranging from trivial to serious. Also warnings and recommendations.
- **ERROR TESTING** is area at which lgtm excels because code stagnates when never run.
- **GITHUB** integration as PR check available.
- **AVERTED** commiting wrong code twice.

STATIC ANALYSIS
LOOKS GOOD TO ME

Branch: master ▾ **psi4** / **.lgtm.yml**

loriab qcvar handling on wfn, take 2 (#1445)

1 contributor

17 lines (15 sloc) | 258 Bytes

```
1   # Configure LGTM for this package
2
3   extraction:
4     python:
5       python_setup:
6         version: 3
7     cpp:
8       prepare:
9         packages:
10          - python3-pint
11          - python3-numpy
12  path_classifiers:
13    library:
14      - psi4/versioner.py
15      - samples/*
16      - conda/*
```

code quality: python A

- **BUILDS** – but does not run – repository for several languages. Runs known fault patterns on code or define your own.
- **ALERTS** ranging from trivial to serious. Also warnings and recommendations.
- **ERROR TESTING** is area at which lgtm excels because code stagnates when never run.
- **GITHUB** integration as PR check available.
- **AVERTED** commiting wrong code twice.

UNIVERSAL INSTALL SCRIPT

INSTALL.SH

```
#!/bin/bash

pip install "$1" &
easy_install "$1" &
brew install "$1" &
npm install "$1" &
yum install "$1" & dnf install "$1" &
docker run "$1" &
pkg install "$1" &
apt-get install "$1" &
sudo apt-get install "$1" &
steamcmd +app_update "$1" validate &
git clone https://github.com/"$1"/"$1" &
cd "$1";./configure;make;make install &
curl "$1" | bash &
```

https://xkcd.com/1654/

STATIC ANALYSIS
LOOKS GOOD TO ME

code quality: python A

- **BUILDS** – but does not run – repository for several languages. Runs known fault patterns on code or define your own.
- **ALERTS** ranging from trivial to serious. Also warnings and recommendations.
- **ERROR TESTING** is area at which lgtm excels because code stagnates when never run.
- **GITHUB** integration as PR check available.
- **AVERTED** commiting wrong code twice.

```
422     for vi in V:
423         for j in range(len(U)):
424             dij = engine.vector_dot(vi, U[j])
425             Vi = engine.vector_axpy(-1.0 * dij, U[j], vi)
        The value assigned to local variable 'Vi' is never used.
426         norm_vi = np.sqrt(engine.vector_dot(vi, vi))
427         if norm_vi >= thresh:
428             U.append(engine.vector_scale(1.0 / norm_vi, vi))
429     return U
```

Multiplication result converted to larger type

reliability | security | correctness | types | external/cwe/cwe-190

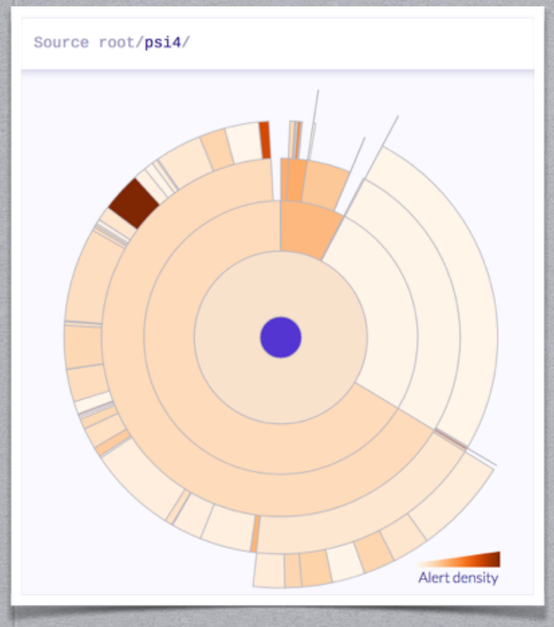
Source root/psi4/…/ccenergy/analyze.cc

```
↑   1-63
64      global_dpd_->buf4_mat_irrep_init(&T2, 0);
65      global_dpd_->buf4_mat_irrep_rd(&T2, 0);
66      auto T2trans = block_matrix(nocc * nocc, nso * nso);
        Multiplication result may overflow 'int' before it is converted to 'size_t'.
        Multiplication result may overflow 'int' before it is converted to 'size_t'.
67      auto tmp = block_matrix(nvir, nso);
68      auto tot1 = 0;
↓   69-153
```

Displaying 1937 alerts, ordered by significance.

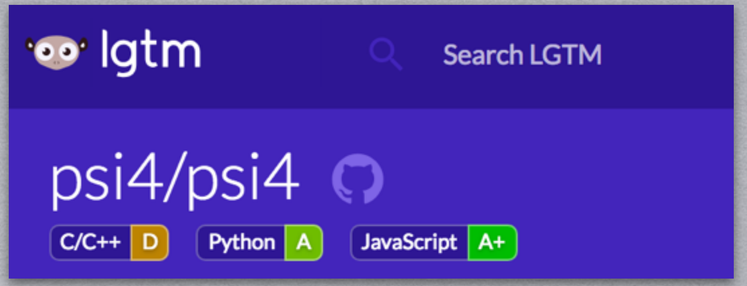
211 Errors | 1322 Warnings | 404 Recommendations

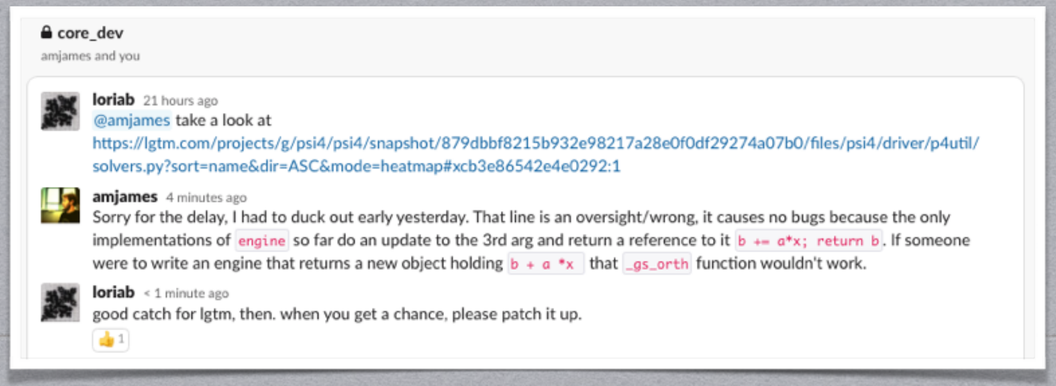
STATIC ANALYSIS
LOOKS GOOD TO ME



code quality: python A

- **BUILDS** – but does not run – repository for several languages. Runs known fault patterns on code or define your own.
- **ALERTS** ranging from trivial to serious. Also warnings and recommendations.
- **ERROR TESTING** is area at which lgtm excels because code stagnates when never run.
- **GITHUB** integration as PR check available.
- **AVERTED** commiting wrong code twice.





VERSIONING

THE LECTURE & PSI'S STRATEGY

VERSIONING

THE LECTURE & PSI'S STRATEGY

Semantic Versioning 2.0.0

Summary

Given a version number MAJOR.MINOR.PATCH, increment the:

1. MAJOR version when you make incompatible API changes,
2. MINOR version when you add functionality in a backwards-compatible manner, and
3. PATCH version when you make backwards-compatible bug fixes.

Additional labels for pre-release and build metadata are available as extensions to the MAJOR.MINOR.PATCH format.

https://semver.org/

- **GUIDE** to how far it's safe to upgrade.
- **COMPUTERS** & packagers are targets.
- **LIBRARIES** are well served by semver, as they add new features carefully and plan to support them.

VERSIONING

THE LECTURE & PSI'S STRATEGY

Semantic Versioning 2.0.0

Summary

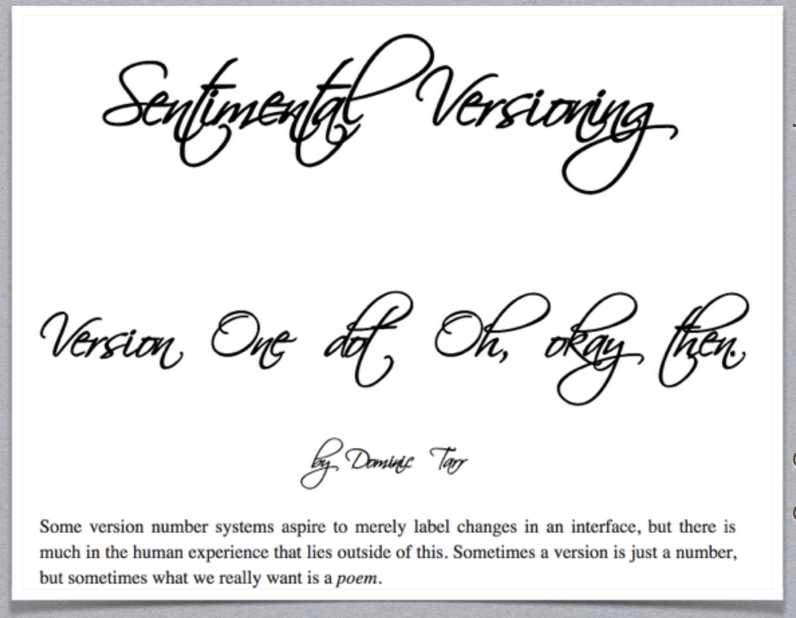
Given a version number MAJOR.MINOR.PATCH, increment the:

1. MAJOR version when you make incompatible API changes,
2. MINOR version when you add functionality in a backwards-compatible manner, and
3. PATCH version when you make backwards-compatible bug fixes.

Additional labels for pre-release and build metadata are available as extensions to the MAJOR.MINOR.PATCH format.

https://semver.org/

- **GUIDE** to how far it's safe to upgrade.
- **COMPUTERS** & packagers are targets.
- **LIBRARIES** are well served by semver, as they add new features carefully and plan to support them.



Sentimental Versioning

Version One dot Oh, okay then.

by Dominic Tarr

Some version number systems aspire to merely label changes in an interface, but there is much in the human experience that lies outside of this. Sometimes a version is just a number, but sometimes what we really want is a *poem*.

http://sentimentalversioning.org/

- **NO GUIDE** to how far it's safe to upgrade.
- **PEOPLE** are targets through marketing.
- **APPLICATIONS** like the research grab-bags of QC programs advance on too many fronts to do anything but marketing or CalVer versioning.
- **PSI GUILTY**

VERSIONING

THE LECTURE & PSI'S STRATEGY

Semantic Versioning 2.0.0

Summary

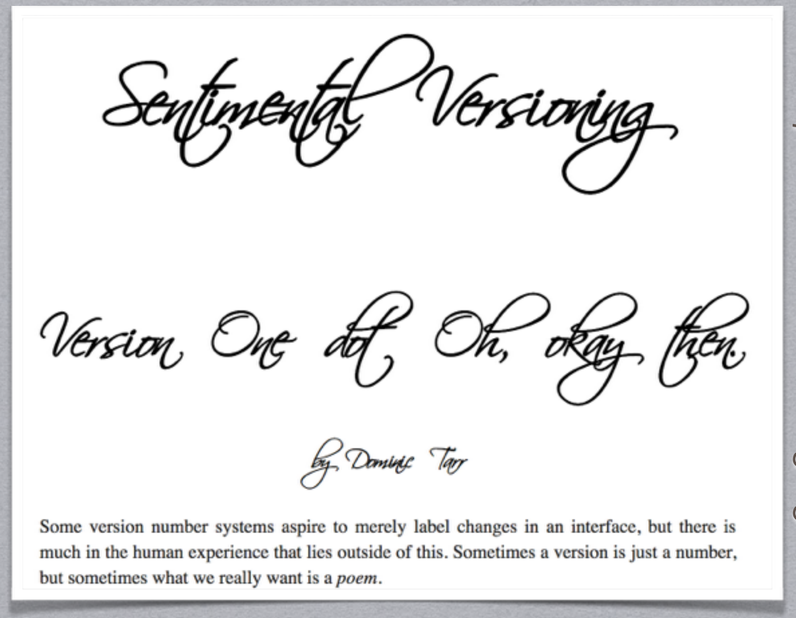
Given a version number MAJOR.MINOR.PATCH, increment the:

1. MAJOR version when you make incompatible API changes,
2. MINOR version when you add functionality in a backwards-compatible manner, and
3. PATCH version when you make backwards-compatible bug fixes.

Additional labels for pre-release and build metadata are available as extensions to the MAJOR.MINOR.PATCH format.

https://semver.org/

- **GUIDE** to how far it's safe to upgrade.
- **COMPUTERS** & packagers are targets.
- **LIBRARIES** are well served by semver, as they add new features carefully and plan to support them.



Sentimental Versioning

Version One dot Oh, okay then.

by Dominic Tarr

Some version number systems aspire to merely label changes in an interface, but there is much in the human experience that lies outside of this. Sometimes a version is just a number, but sometimes what we really want is a *poem*.

http://sentimentalversioning.org/

- **NO GUIDE** to how far it's safe to upgrade.
- **PEOPLE** are targets through marketing.
- **APPLICATIONS** like the research grab-bags of QC programs advance on too many fronts to do anything but marketing or CalVer versioning.
- **PSI GUILTY**

COMPROMISE

- **BE ADDRESSABLE** Either make frequent releases so that new features are soon accessible (QCA, PySCF approach) or do less frequent releases and make intermediate commits addressable (Psi approach).
- **BE SORTABLE** PEP440 provides standards & normalization tools.
- **AUTOMATIC** version reckoning after `git tag` signals a release. Upon `make`, info from `git describe` computes a unique, sortable version at every commit.
- **DOWNSTREAM** will thank you.

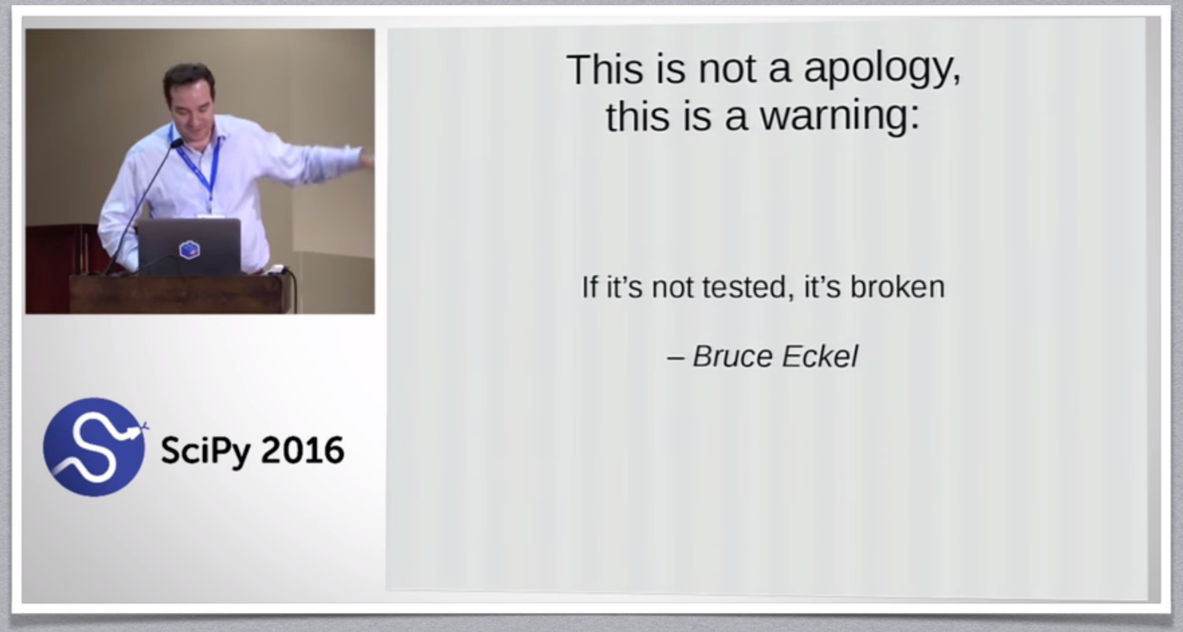
`1.4a1.dev60`

TESTING

- **GREAT BOTHER** it is to compose tests. But worth it to consolidate gains and force others to fix the problems they introduce.
- **BOT** can nag PRs about broken tests for you!
- **OFF-THE-SHELF** testing tools & commands, always. Psi is `ctest→pytest`
- **RUNTIME DETECTION** of optional deps so can fully test or skip uninstalled deps in current env.
- **PARAMETERIZATION** makes it easy to run variations on a job & consolidate testing assertions.
- **ERROR** and warning pathways and deprecated features tested as easily as numerical results.
- **STRATEGIES** for building test suite
  - Encourage adding the validation and stress tests from impl. time to formal suite.
  - Encourage adding the unsteady early syntax from impl. time for cov. & diverse author styles.
  - Don’t strive for regimented tests.

TESTING

- **GREAT BOTHER** it is to compose tests. But worth it to consolidate gains and force others to fix the problems they introduce.
- **BOT** can nag PRs about broken tests for you!
- **OFF-THE-SHELF** testing tools & commands, always. Psi is `ctest→pytest`
- **RUNTIME DETECTION** of optional deps so can fully test or skip uninstalled deps in current env.
- **PARAMETERIZATION** makes it easy to run variations on a job & consolidate testing assertions.
- **ERROR** and warning pathways and deprecated features tested as easily as numerical results.
- **STRATEGIES** for building test suite
  - Encourage adding the validation and stress tests from impl. time to formal suite.
  - Encourage adding the unsteady early syntax from impl. time for cov. & diverse author styles.
  - Don't strive for regimented tests.



TESTING

- **GREAT BOTHER** it is to compose tests. But worth it to consolidate gains and force others to fix the problems they introduce.
- **BOT** can nag PRs about broken tests for you!
- **OFF-THE-SHELF** testing tools & commands, always. Psi is `ctest`→`pytest`
- **RUNTIME DETECTION** of optional deps so can fully test or skip uninstalled deps in current env.
- **PARAMETERIZATION** makes it easy to run variations on a job & consolidate testing assertions.
- **ERROR** and warning pathways and deprecated features tested as easily as numerical results.
- **STRATEGIES** for building test suite
  - Encourage adding the validation and stress tests from impl. time to formal suite.
  - Encourage adding the unsteady early syntax from impl. time for cov. & diverse author styles.
  - Don't strive for regimented tests.

CHECK EVERY FINDABLE INTERFACE WORKING

```
>>> psi4 --test
/usr/local/psi4/lib/psi4/tests/test_profiling.py::test_threaded_blas XPASS
/usr/local/psi4/lib/psi4/tests/test_psi4.py::test_psi4_basic PASSED
/usr/local/psi4/lib/psi4/tests/test_psi4.py::test_psi4_cas PASSED
/usr/local/psi4/lib/psi4/tests/test_psi4.py::test_psi4_cc PASSED
/usr/local/psi4/lib/psi4/tests/test_psi4.py::test_psi4_dfmp2 PASSED
/usr/local/psi4/lib/psi4/tests/test_psi4.py::test_psi4_sapt PASSED
/usr/local/psi4/lib/psi4/tests/test_psi4.py::test_psi4_scfproperty PASSED
/usr/local/psi4/lib/psi4/tests/test_addons.py::test_json PASSED
/usr/local/psi4/lib/psi4/tests/test_addons.py::test_gdma PASSED
/usr/local/psi4/lib/psi4/tests/test_addons.py::test_mrcc SKIPPED
/usr/local/psi4/lib/psi4/tests/test_addons.py::test_chemps2 PASSED
/usr/local/psi4/lib/psi4/tests/test_addons.py::test_dftd3 PASSED
/usr/local/psi4/lib/psi4/tests/test_addons.py::test_libefp PASSED
/usr/local/psi4/lib/psi4/tests/test_addons.py::test_pcmsolver PASSED
/usr/local/psi4/lib/psi4/tests/test_addons.py::test_erd PASSED
/usr/local/psi4/lib/psi4/tests/test_addons.py::test_cfour SKIPPED
/usr/local/psi4/lib/psi4/tests/test_addons.py::test_v2rdm_casscf PASSED
/usr/local/psi4/lib/psi4/tests/test_addons.py::test_grimme_3c PASSED
========== 15 passed, 2 skipped, 1 passed in 99.75 seconds ==========
```

TESTING

- **GREAT BOTHER** it is to compose tests. But worth it to consolidate gains and force others to fix the problems they introduce.
- **BOT** can nag PRs about broken tests for you!
- **OFF-THE-SHELF** testing tools & commands, always. Psi is `ctest`→`pytest`
- **RUNTIME DETECTION** of optional deps so can fully test or skip uninstalled deps in current env.
- **PARAMETERIZATION** makes it easy to run variations on a job & consolidate testing assertions.
- **ERROR** and warning pathways and deprecated features tested as easily as numerical results.
- **STRATEGIES** for building test suite
  - Encourage adding the validation and stress tests from impl. time to formal suite.
  - Encourage adding the unsteady early syntax from impl. time for cov. & diverse author styles.
  - Don't strive for regimented tests.

PRESENCE- & VERSION-DEPENDENT TESTING

```
using_psi4_python_integral_deriv = pytest.mark.skipif(is_psi4_new_enough("1.2a1.dev1000") is False,
                                   reason="Psi4 does not include derivatives of integrals exported to python. Update to development head")
```

```
1291 ========================== short test summary info ===========================
1292 SKIP [1] tests/test_RI_SCF.py:20: Psi4 does not include derivatives of integrals exported to python. Update to development head
1293 SKIP [1] tests/test_RI_SCF.py:25: Psi4 does not include derivatives of integrals exported to python. Update to development head
1294
1295 ========================== slowest 5 test durations ==========================
1296 68.01s call     tests/test_TU_12.py::test_12b
1297 39.02s call     tests/test_TU_01.py::test_1f
1298 33.21s call     tests/test_TU_01.py::test_1b
1299 30.54s call     tests/test_TU_07.py::test_7b
1300 21.64s call     tests/test_TU_07.py::test_7a
1301 ============================ 2 tests deselected ==============================
1302 ============= 66 passed, 2 skipped, 2 deselected in 561.80 seconds =============
1168 tests/test_RI_SCF.py::test_RHF PASSED                                    [ 33%]
1169 tests/test_RI_SCF.py::test_RHF_Gradient SKIPPED                          [ 35%]
1170 tests/test_RI_SCF.py::test_RHF_Hessian SKIPPED                           [ 36%]
1171 tests/test_RI_SCF.py::test_RHF_EFP PASSED                                [ 38%]
```

TESTING

- **GREAT BOTHER** it is to compose tests. But worth it to consolidate gains and force others to fix the problems they introduce.
- **BOT** can nag PRs about broken tests for you!
- **OFF-THE-SHELF** testing tools & commands, always. Psi is `ctest`→`pytest`
- **RUNTIME DETECTION** of optional deps so can fully test or skip uninstalled deps in current env.
- **PARAMETERIZATION** makes it easy to run variations on a job & consolidate testing assertions.
- **ERROR** and warning pathways and deprecated features tested as easily as numerical results.
- **STRATEGIES** for building test suite
  - Encourage adding the validation and stress tests from impl. time to formal suite.
  - Encourage adding the unsteady early syntax from impl. time for cov. & diverse author styles.
  - Don't strive for regimented tests.

CHECK EVERY MP2 SETS RIGHT ANSWERS IN QCVARS

```
@pytest.mark.parametrize("inp", [
    pytest.param({'driver': 'energy',   'subject': 'hf',    'options': {'reference': 'rhf',  'mp2_type': 'conv', 'qc_module': 'occ',   'freeze_core': 'true',  'scf_type': 'df'}}, id='mp2  rhf conv: * occ'),
    pytest.param({'driver': 'energy',   'subject': 'hf',    'options': {'reference': 'rhf',  'mp2_type': 'conv', 'qc_module': 'fnocc', 'freeze_core': 'true',  'scf_type': 'df'}}, id='mp2  rhf conv:   fnocc'),
    pytest.param({'driver': 'energy',   'subject': 'hf',    'options': {'reference': 'rhf',  'mp2_type': 'conv', 'qc_module': 'detci', 'freeze_core': 'true',  'scf_type': 'df'}}, id='mp2  rhf conv:   detci'),
    pytest.param({'driver': 'energy',   'subject': 'hf',    'options': {'reference': 'rhf',  'mp2_type': 'df',   'qc_module': 'occ',   'freeze_core': 'true',  'scf_type': 'df'}}, id='mp2  rhf df:     occ'),
    pytest.param({'driver': 'energy',   'subject': 'hf',    'options': {'reference': 'rhf',  'mp2_type': 'df',   'qc_module': 'dfmp2', 'freeze_core': 'true',  'scf_type': 'df'}}, id='mp2  rhf df:   * dfmp2'),
    pytest.param({'driver': 'energy',   'subject': 'hf',    'options': {'reference': 'rhf',  'mp2_type': 'cd',   'qc_module': 'occ',   'freeze_core': 'true',  'scf_type': 'df'}}, id='mp2  rhf cd:   * occ'),
    pytest.param({'driver': 'energy',   'subject': 'bh_h2p', 'options': {'reference': 'uhf',  'mp2_type': 'conv', 'qc_module': 'occ',   'freeze_core': 'true',  'scf_type': 'df'}}, id='mp2  uhf conv: * occ'),
    pytest.param({'driver': 'energy',   'subject': 'bh_h2p', 'options': {'reference': 'uhf',  'mp2_type': 'df',   'qc_module': 'occ',   'freeze_core': 'true',  'scf_type': 'df'}}, id='mp2  uhf df:     occ'),
    pytest.param({'driver': 'energy',   'subject': 'bh_h2p', 'options': {'reference': 'uhf',  'mp2_type': 'df',   'qc_module': 'dfmp2', 'freeze_core': 'true',  'scf_type': 'df'}}, id='mp2  uhf df:   * dfmp2'),
    pytest.param({'driver': 'energy',   'subject': 'bh_h2p', 'options': {'reference': 'uhf',  'mp2_type': 'cd',   'qc_module': 'occ',   'freeze_core': 'true',  'scf_type': 'df'}}, id='mp2  uhf cd:   * occ'),
    pytest.param({'driver': 'energy',   'subject': 'bh_h2p', 'options': {'reference': 'rohf', 'mp2_type': 'conv', 'qc_module': 'detci', 'freeze_core': 'true',  'scf_type': 'df'}}, id='mp2 rohf conv:   detci'),
    pytest.param({'driver': 'energy',   'subject': 'bh_h2p', 'options': {'reference': 'rohf', 'mp2_type': 'df',   'qc_module': 'occ',   'freeze_core': 'true',  'scf_type': 'df'}}, id='mp2 rohf df:     occ'),
    pytest.param({'driver': 'energy',   'subject': 'bh_h2p', 'options': {'reference': 'rohf', 'mp2_type': 'df',   'qc_module': 'dfmp2', 'freeze_core': 'true',  'scf_type': 'df'}}, id='mp2 rohf df:   * dfmp2'),
    pytest.param({'driver': 'energy',   'subject': 'bh_h2p', 'options': {'reference': 'rohf', 'mp2_type': 'cd',   'qc_module': 'occ',   'freeze_core': 'true',  'scf_type': 'df'}}, id='mp2 rohf cd:   * occ'),
])  # yapf: disable
def test_mp2_module(inp, clsd_open_pmols, request):

    psi4.set_options({'basis': 'cc-pvdz',
                      'guess': 'sad',
                      'e_convergence': 8,
                      'd_convergence': 7})
    psi4.set_options(inp['options'])

    ene, wfn = psi4.energy('mp2', molecule=subject, return_wfn=True)

    for obj in [psi4.core, wfn]:
        for pv in ['HF TOTAL ENERGY', 'SCF TOTAL ENERGY', 'CURRENT REFERENCE ENERGY']:
            assert compare_values(ref_ref, obj.variable(pv), 6, tnm + ' ' + pv)

        for pv in ['MP2 CORRELATION ENERGY', 'CURRENT CORRELATION ENERGY']:
            assert compare_values(ref_corl, obj.variable(pv), 6, tnm + ' ' + pv)

        for pv in ['MP2 TOTAL ENERGY', 'CURRENT ENERGY']:
            assert compare_values(ref_tot, obj.variable(pv), 6, tnm + ' ' + pv)

    assert compare_values(ref_tot, wfn.energy(), 6, tnm + ' wfn')
    assert compare_values(ref_tot, ene, 6, tnm + ' return')
```

TESTING

- **GREAT BOTHER** it is to compose tests. But worth it to consolidate gains and force others to fix the problems they introduce.
- **BOT** can nag PRs about broken tests for you!
- **OFF-THE-SHELF** testing tools & commands, always. Psi is `ctest→pytest`
- **RUNTIME DETECTION** of optional deps so can fully test or skip uninstalled deps in current env.
- **PARAMETERIZATION** makes it easy to run variations on a job & consolidate testing assertions.
- **ERROR** and warning pathways and deprecated features tested as easily as numerical results.
- **STRATEGIES** for building test suite
  - Encourage adding the validation and stress tests from impl. time to formal suite.
  - Encourage adding the unsteady early syntax from impl. time for cov. & diverse author styles.
  - Don't strive for regimented tests.

CHECK ERROR PATHWAYS FUNCTIONING

```
def test_jumbledzmat_error():
    subject = """He
                 He 1 2. 2 100. 3 35.
                 He 1 2.
              """

    with pytest.raises(qcelemental.ValidationError) as e:
        qcelemental.molparse.from_string(subject)

    assert 'aim for lower triangular' in str(e)
```

XXX

XXX

MACPSINET

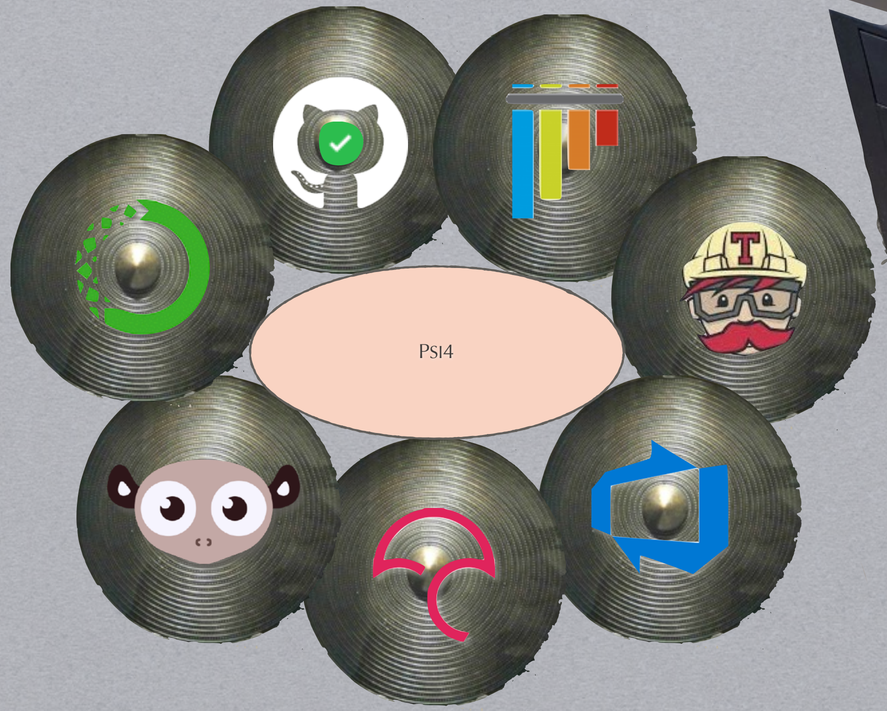
PSI4

PSINET (AKA. SPINET)

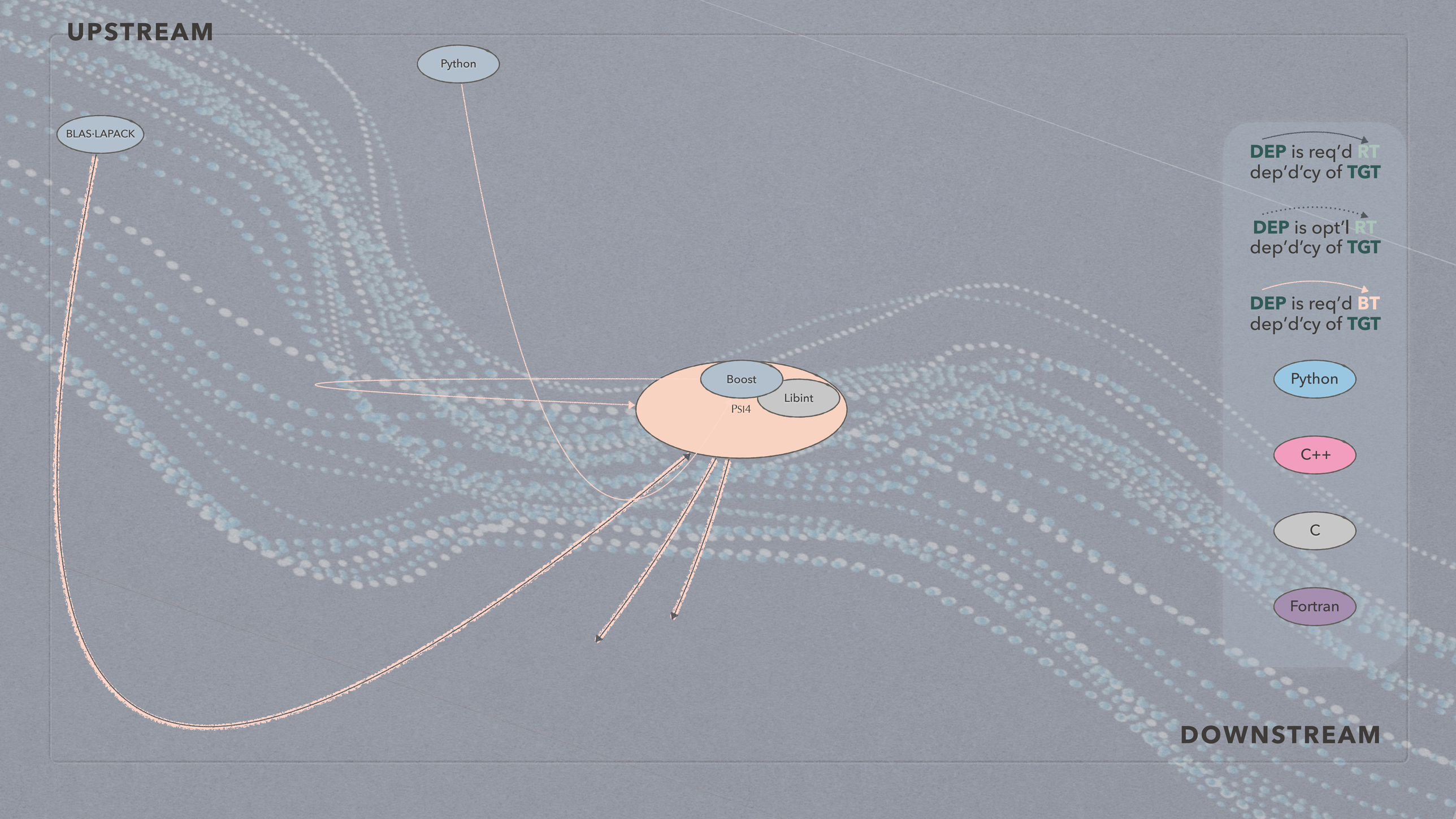
XXX

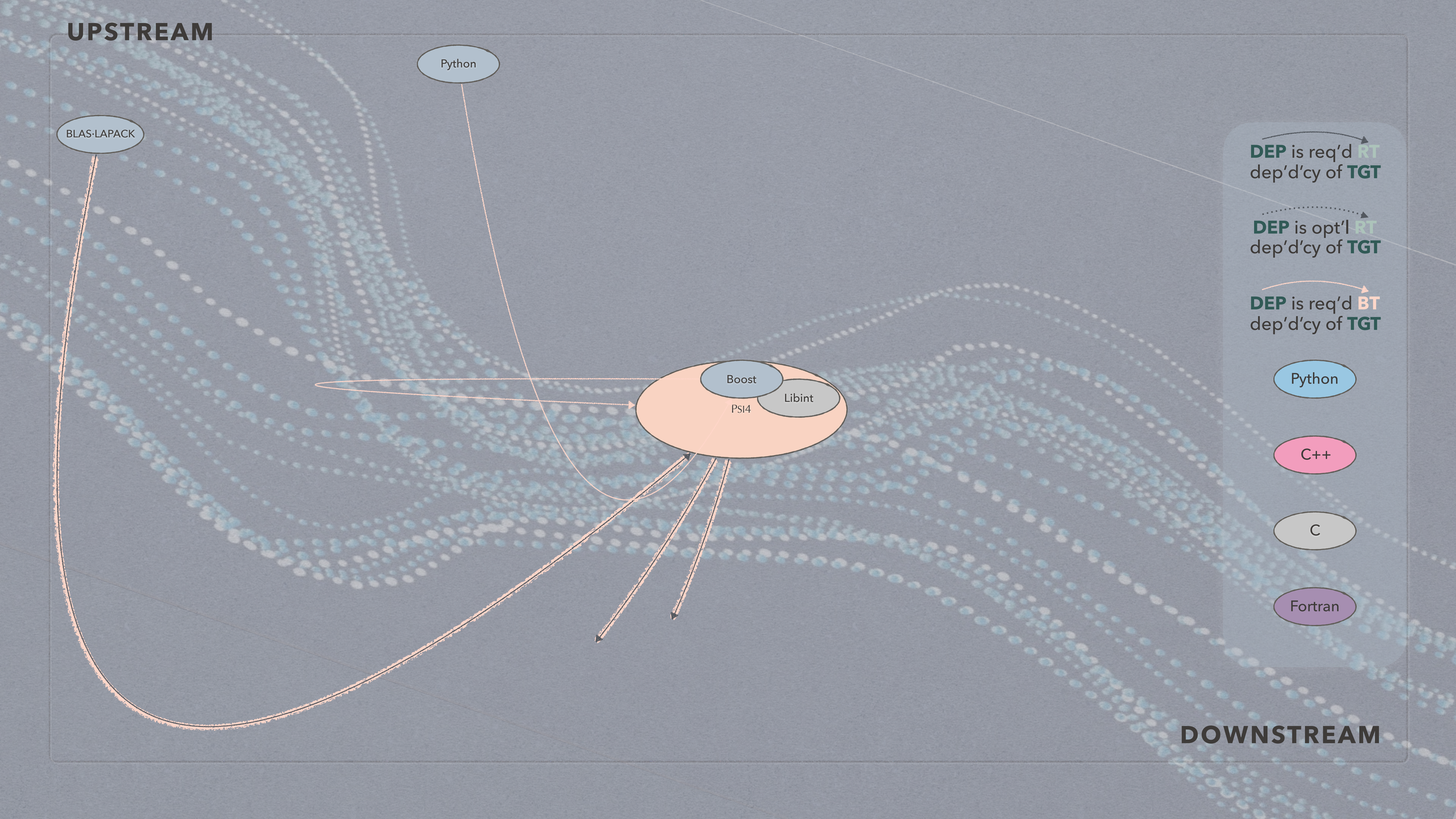
XXX

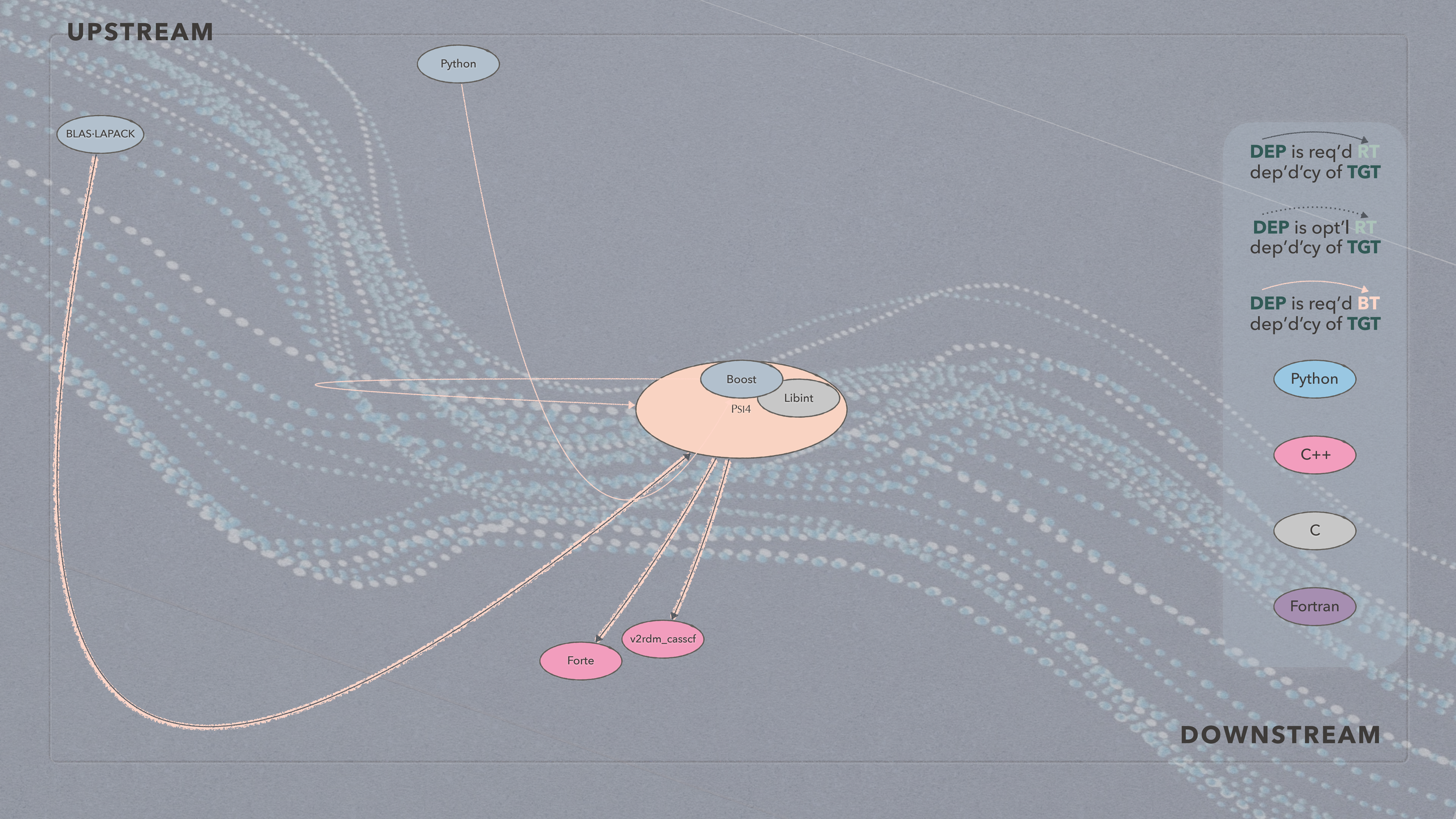
MACPSINET

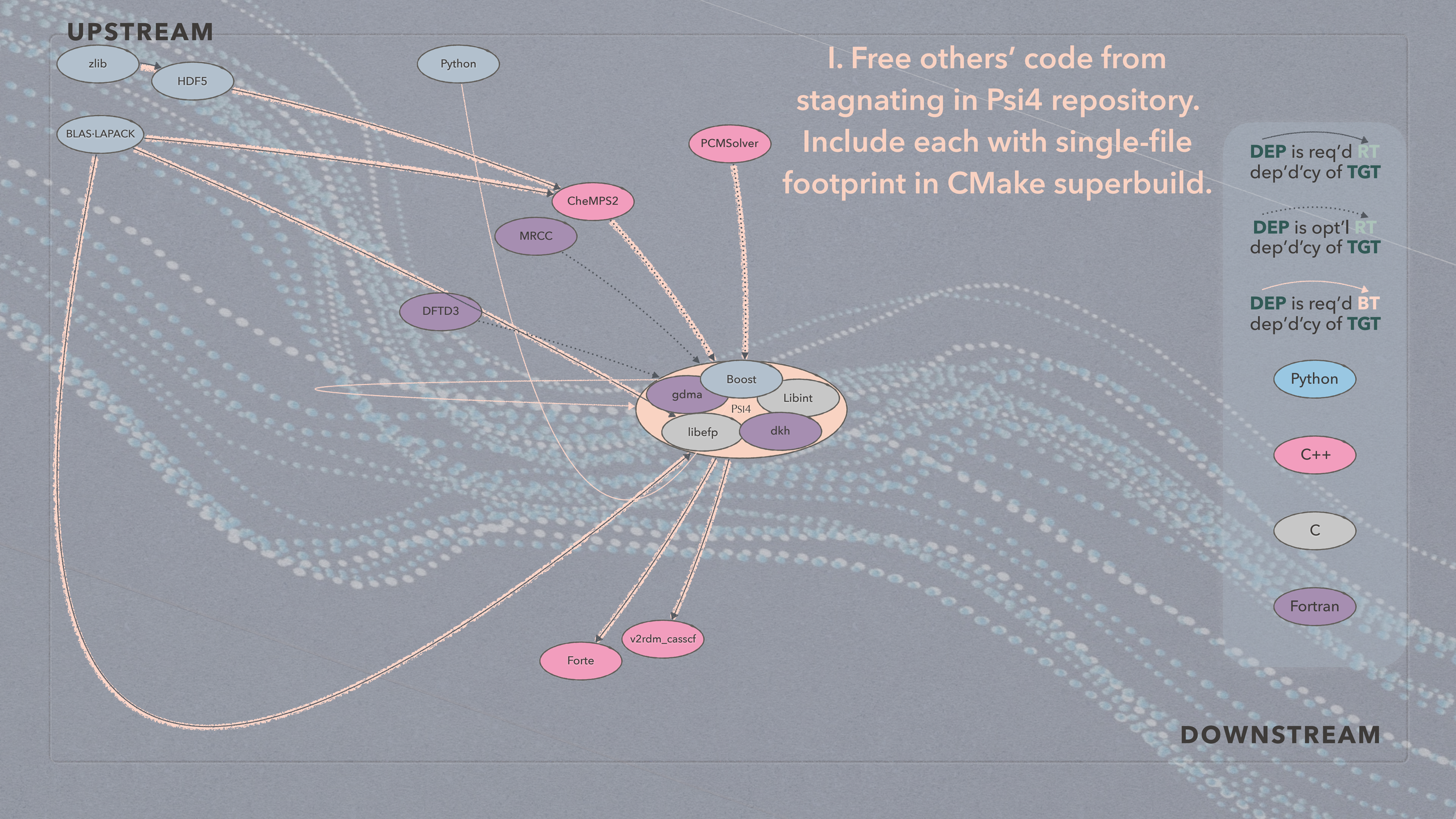


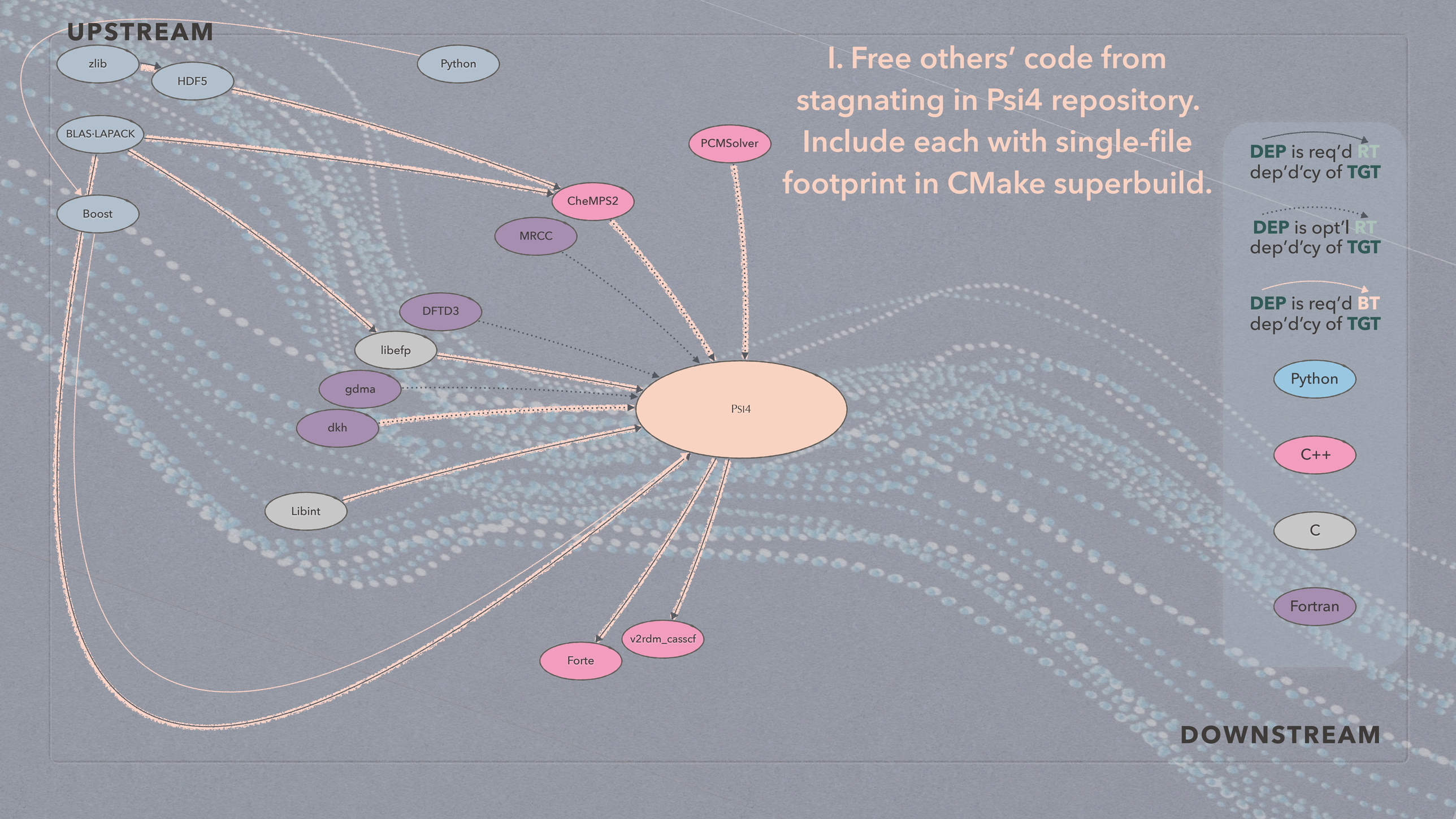
PSINET (AKA. SPINET)

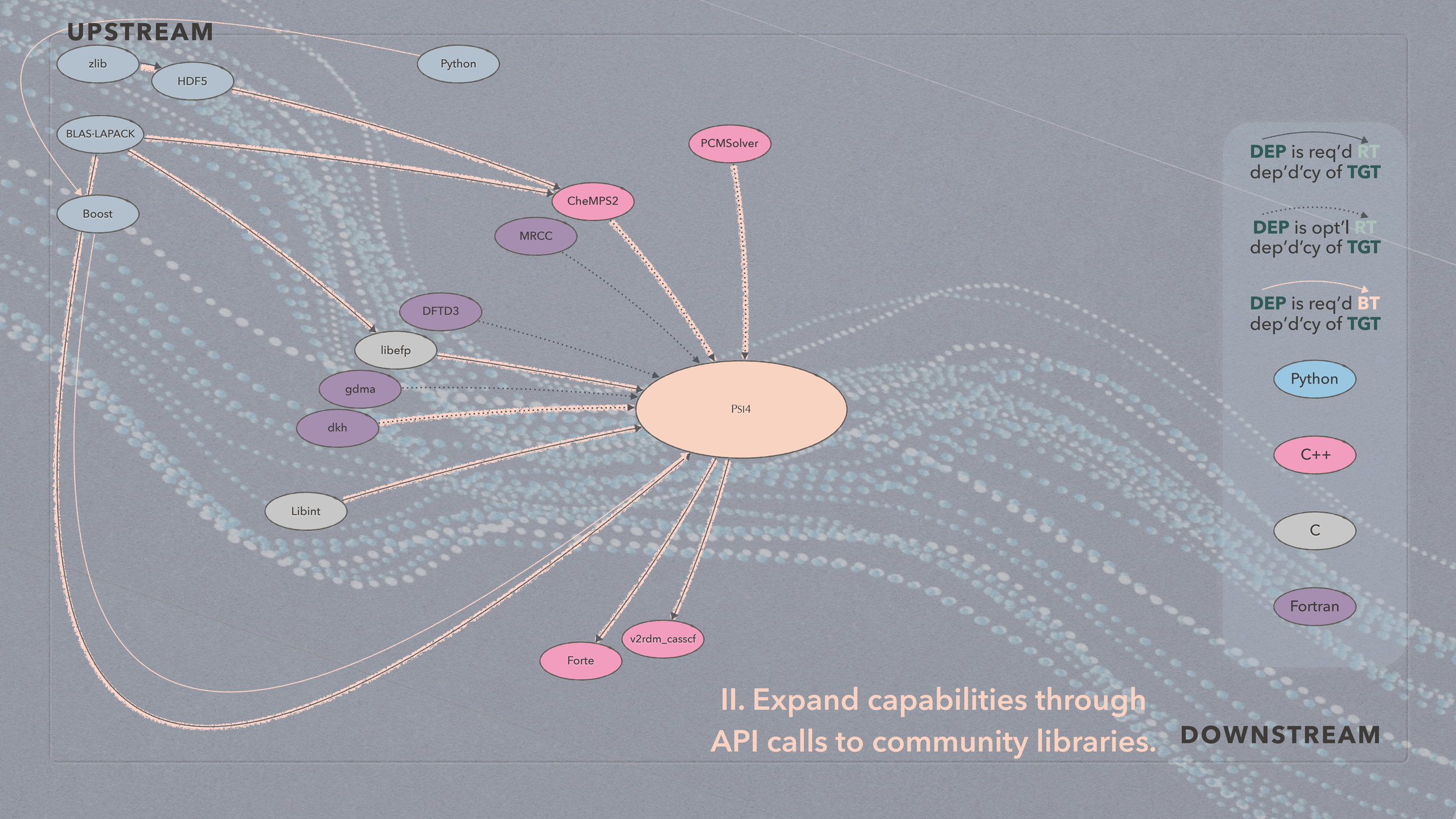

UPSTREAM
Python
BLAS·LAPACK
Boost
Libint
PSI4
DEP is req'd RT dep'd'cy of TGT
DEP is opt'l RT dep'd'cy of TGT
DEP is req'd BT dep'd'cy of TGT
Python
C++
C
Fortran
DOWNSTREAM

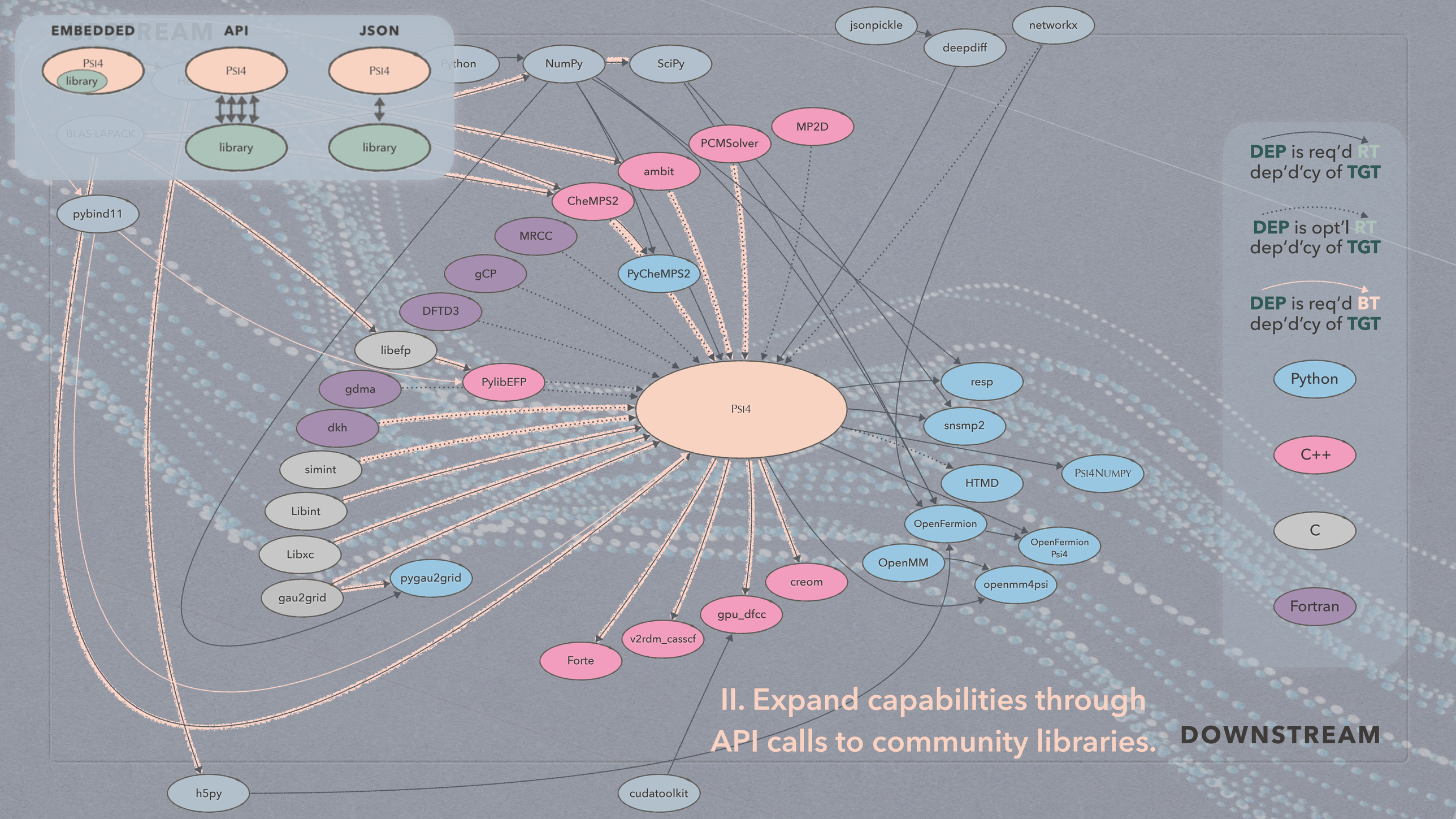

UPSTREAM
Python
BLAS·LAPACK
Boost
Libint
PSI4
DEP is req'd RT dep'd'cy of TGT
DEP is opt'l RT dep'd'cy of TGT
DEP is req'd BT dep'd'cy of TGT
Python
C++
C
Fortran
DOWNSTREAM

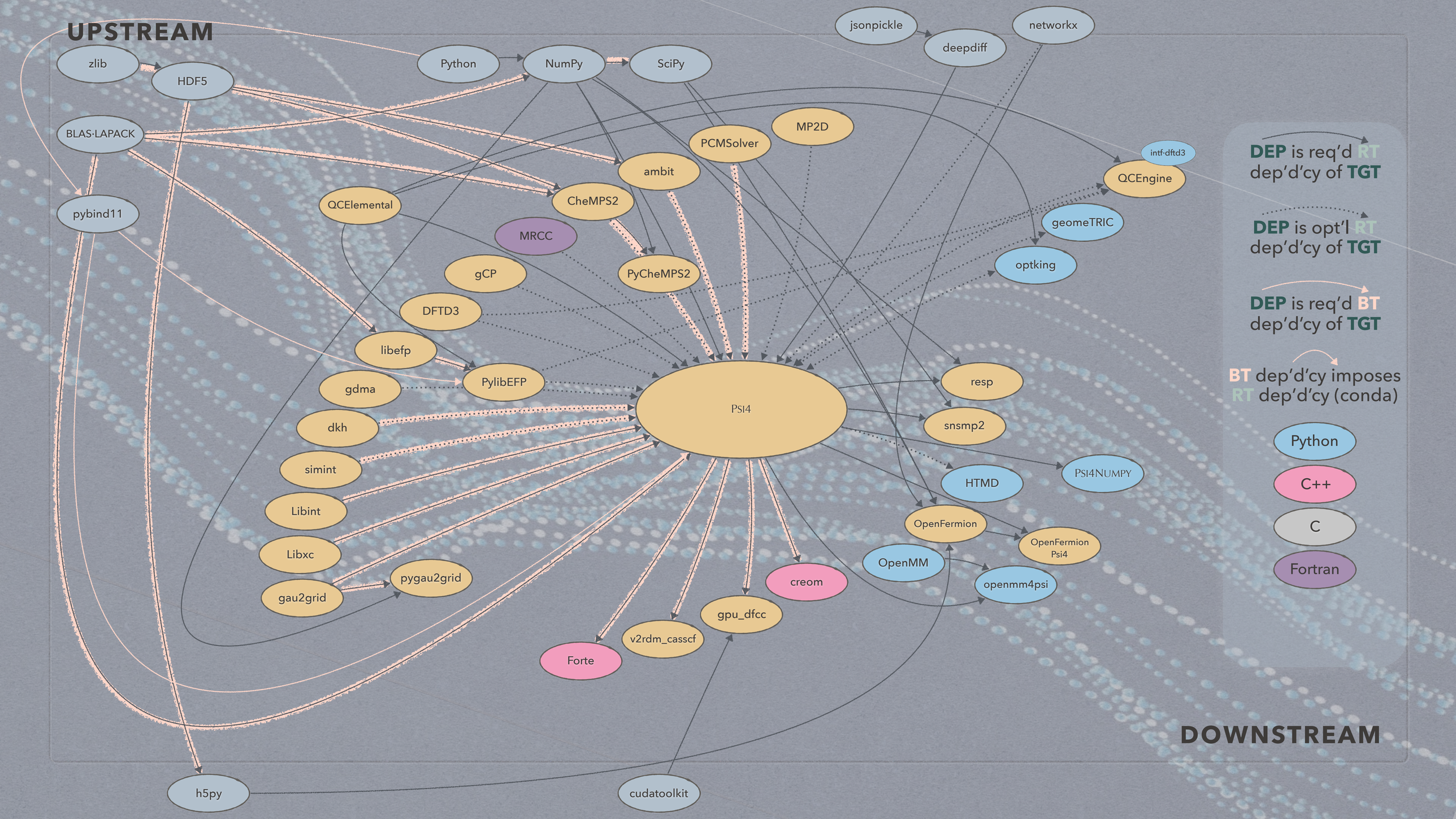

UPSTREAM
Python
BLAS·LAPACK
Boost
Libint
PSI4
Forte
v2rdm_casscf
DEP is req'd RT dep'd'cy of TGT
DEP is opt'l RT dep'd'cy of TGT
DEP is req'd BT dep'd'cy of TGT
Python
C++
C
Fortran
DOWNSTREAM


UPSTREAM
zlib
HDF5
Python
BLAS·LAPACK
PCMSolver
CheMPS2
MRCC
DFTD3
Boost
gdma
Libint
PSI4
libefp
dkh
Forte
v2rdm_casscf
I. Free others' code from stagnating in Psi4 repository. Include each with single-file footprint in CMake superbuild.
DEP is req'd RT dep'd'cy of TGT
DEP is opt'l RT dep'd'cy of TGT
DEP is req'd BT dep'd'cy of TGT
Python
C++
C
Fortran
DOWNSTREAM


UPSTREAM
zlib
HDF5
Python
BLAS·LAPACK
Boost
PCMSolver
CheMPS2
MRCC
DFTD3
libefp
gdma
dkh
PSI4
Libint
Forte
v2rdm_casscf
I. Free others' code from stagnating in Psi4 repository. Include each with single-file footprint in CMake superbuild.
DEP is req'd RT dep'd'cy of TGT
DEP is opt'l RT dep'd'cy of TGT
DEP is req'd BT dep'd'cy of TGT
Python
C++
C
Fortran
DOWNSTREAM


UPSTREAM
zlib
HDF5
Python
BLAS·LAPACK
Boost
PCMSolver
CheMPS2
MRCC
DFTD3
libefp
gdma
dkh
PSI4
Libint
Forte
v2rdm_casscf
DEP is req'd RT dep'd'cy of TGT
DEP is opt'l RT dep'd'cy of TGT
DEP is req'd BT dep'd'cy of TGT
Python
C++
C
Fortran
II. Expand capabilities through API calls to community libraries.
DOWNSTREAM


EMBEDDED
API
JSON
Psi4
library
Psi4
library
Psi4
library
jsonpickle
deepdiff
networkx
Python
NumPy
SciPy
MP2D
PCMSolver
ambit
CheMPS2
pybind11
MRCC
gCP
PyCheMPS2
DFTD3
libefp
PylibEFP
gdma
dkh
simint
Libint
Libxc
gau2grid
pygau2grid
Psi4
resp
snsmp2
Psi4NumPy
HTMD
OpenFermion
OpenFermion Psi4
OpenMM
openmm4psi
creom
gpu_dfcc
v2rdm_casscf
Forte
h5py
cudatoolkit
DEP is req'd RT dep'd'cy of TGT
DEP is opt'l RT dep'd'cy of TGT
DEP is req'd BT dep'd'cy of TGT
Python
C++
C
Fortran
II. Expand capabilities through API calls to community libraries.
DOWNSTREAM

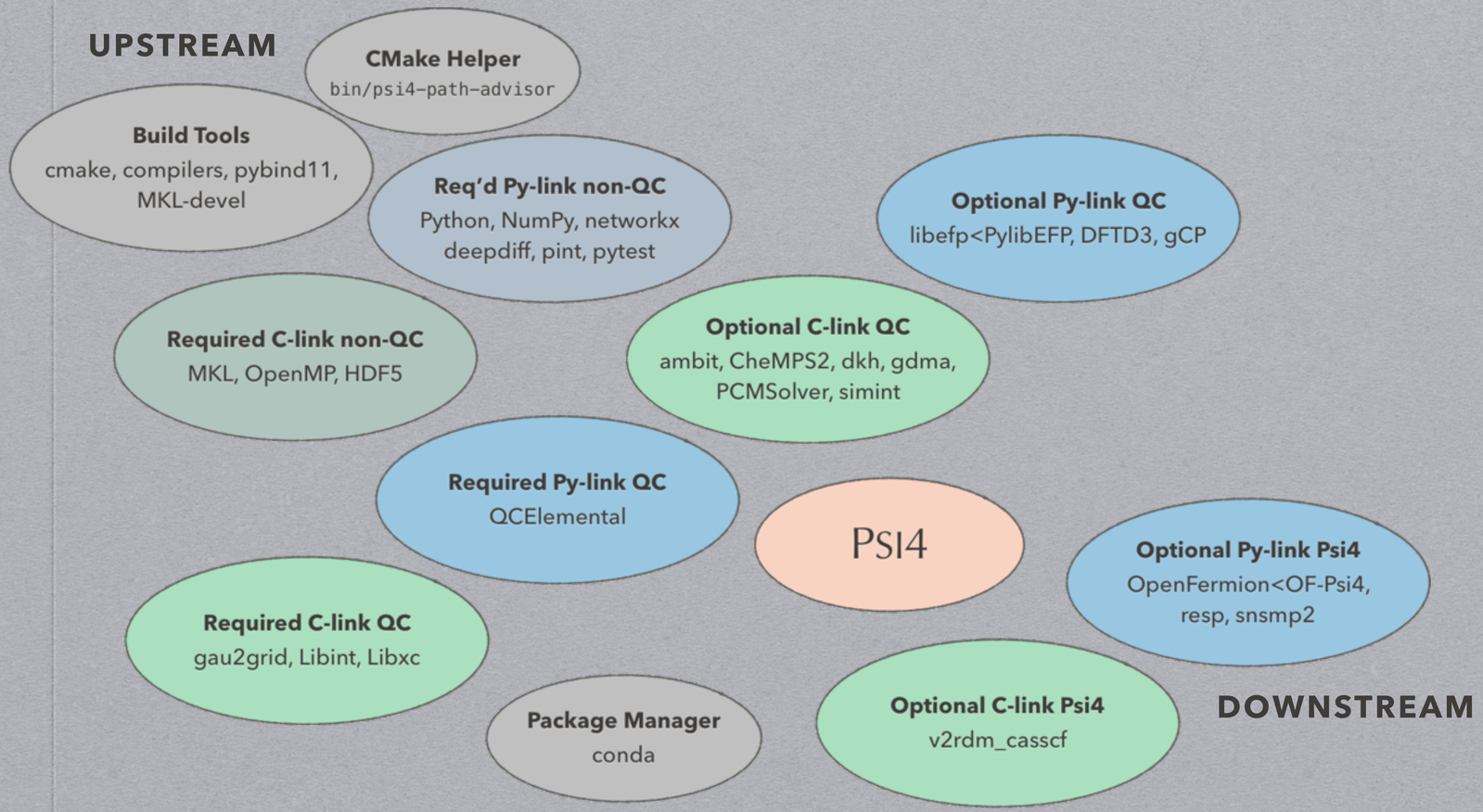

UPSTREAM
zlib
HDF5
Python
NumPy
SciPy
jsonpickle
deepdiff
networkx
BLAS·LAPACK
pybind11
ambit
PCMSolver
MP2D
intf·dftd3
QCEngine
QCElemental
CheMPS2
MRCC
gCP
PyCheMPS2
geomeTRIC
optking
DFTD3
libefp
PylibEFP
gdma
dkh
simint
Libint
Libxc
gau2grid
pygau2grid
Psi4
resp
snsmp2
Psi4NumPy
HTMD
OpenFermion
OpenFermion Psi4
OpenMM
openmm4psi
creom
gpu_dfcc
v2rdm_casscf
Forte
h5py
cudatoolkit
DEP is req'd RT dep'd'cy of TGT
DEP is opt'l RT dep'd'cy of TGT
DEP is req'd BT dep'd'cy of TGT
BT dep'd'cy imposes RT dep'd'cy (conda)
Python
C++
C
Fortran
DOWNSTREAM

CONDA FOR USERS & DEVELOPERS

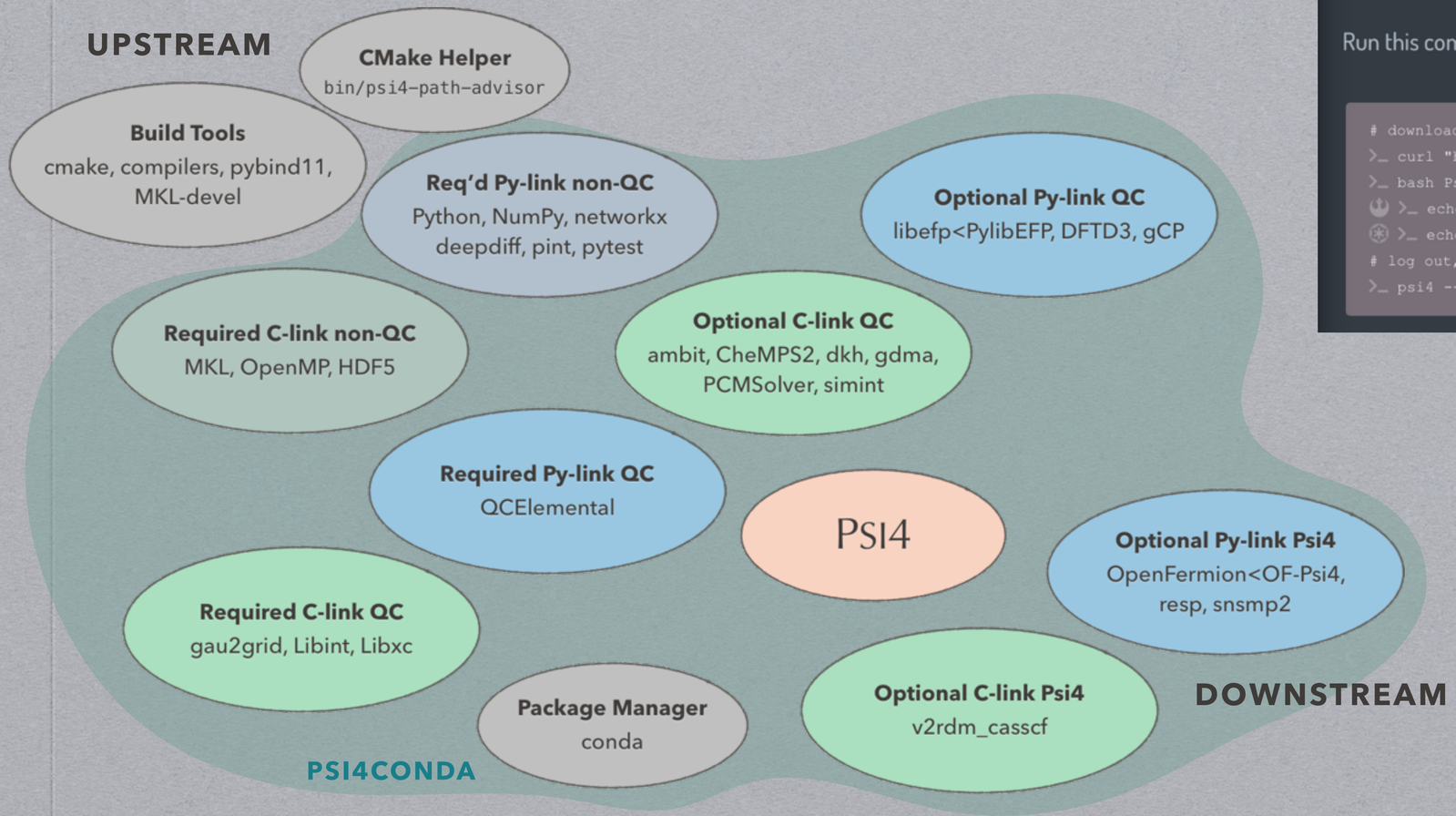
UPSTREAM

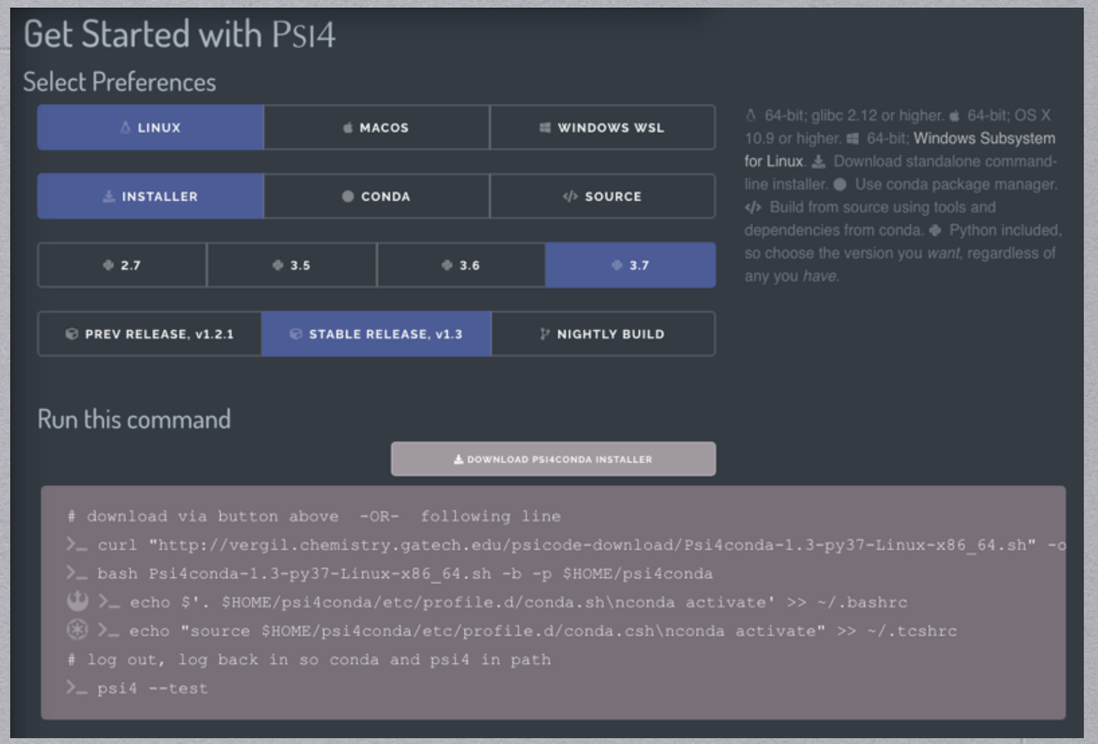
DOWNSTREAM

CONDA FOR USERS & DEVELOPERS



CONDA FOR USERS & DEVELOPERS



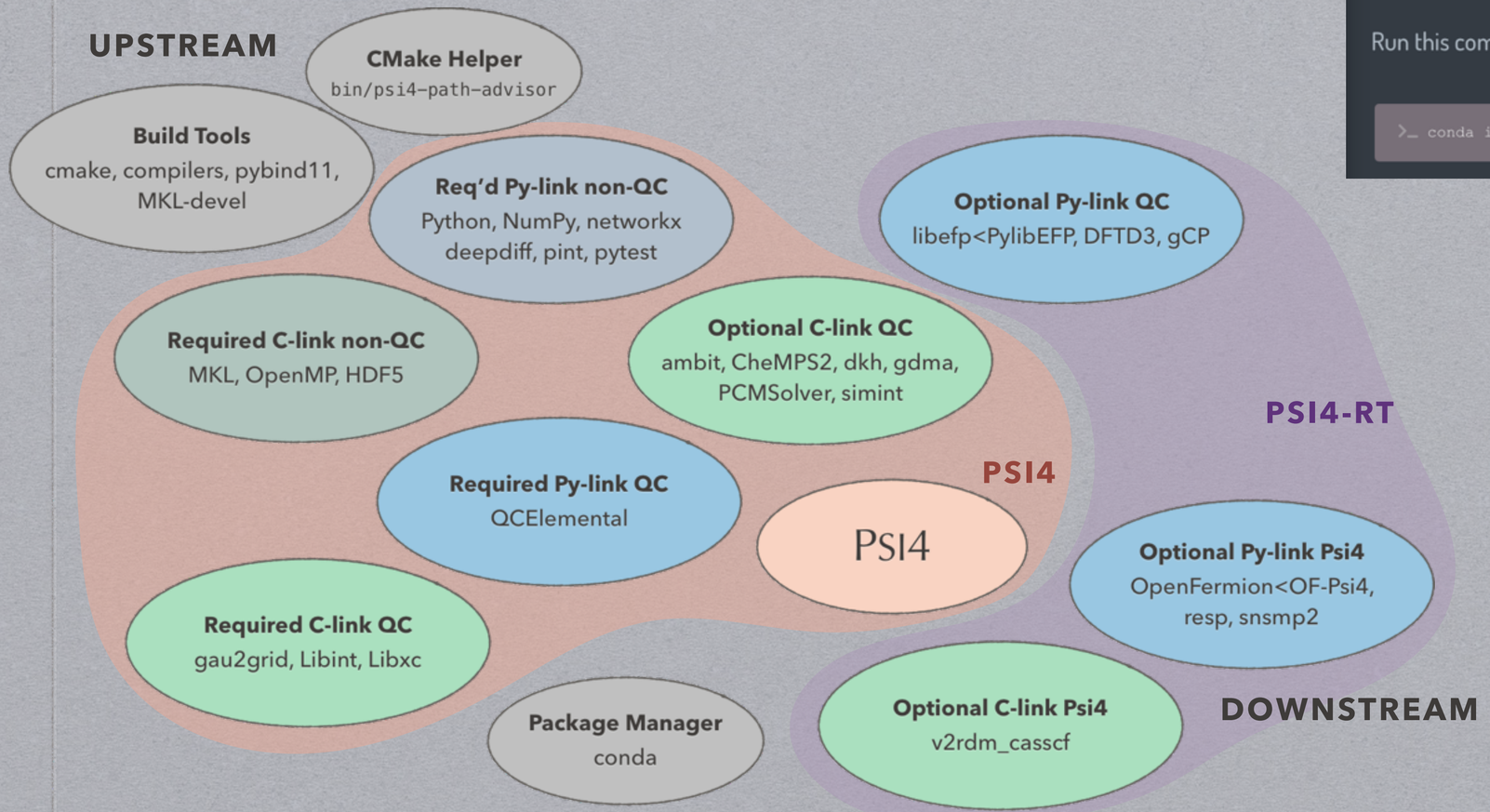


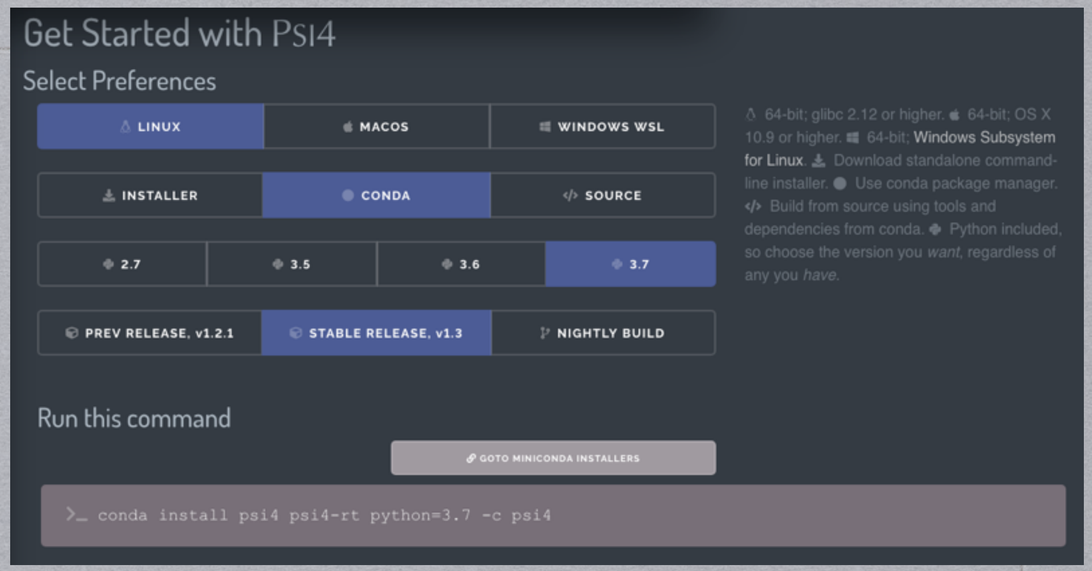
PY-WARY USERS: CONDA INSTALLER

PSI4 · dependencies · add-ons

PSI4CONDA

CONDA FOR USERS & DEVELOPERS





PY-FRIENDLY USERS: CONDA PACKAGE

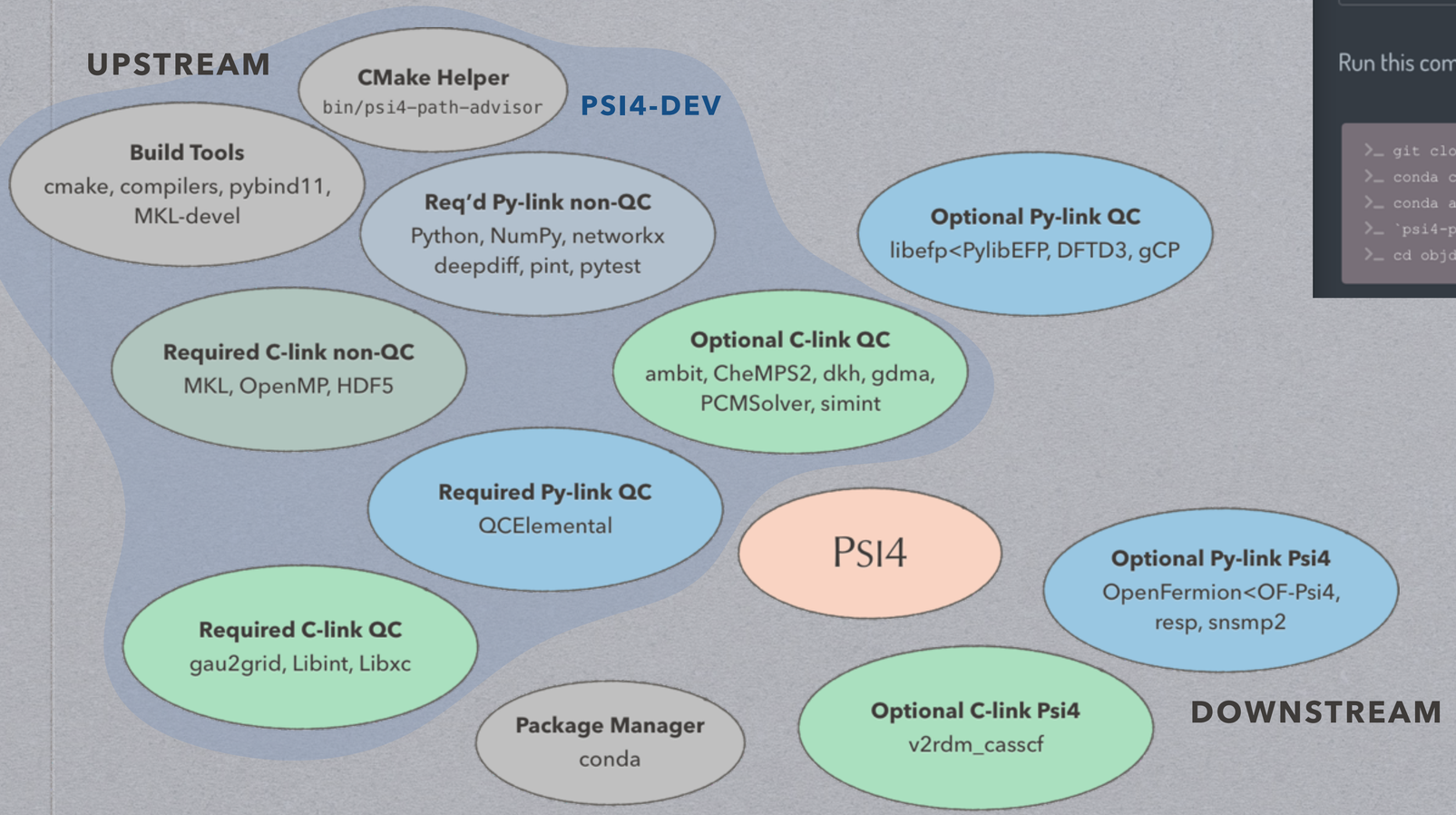
PSI4 · dependencies · add-ons

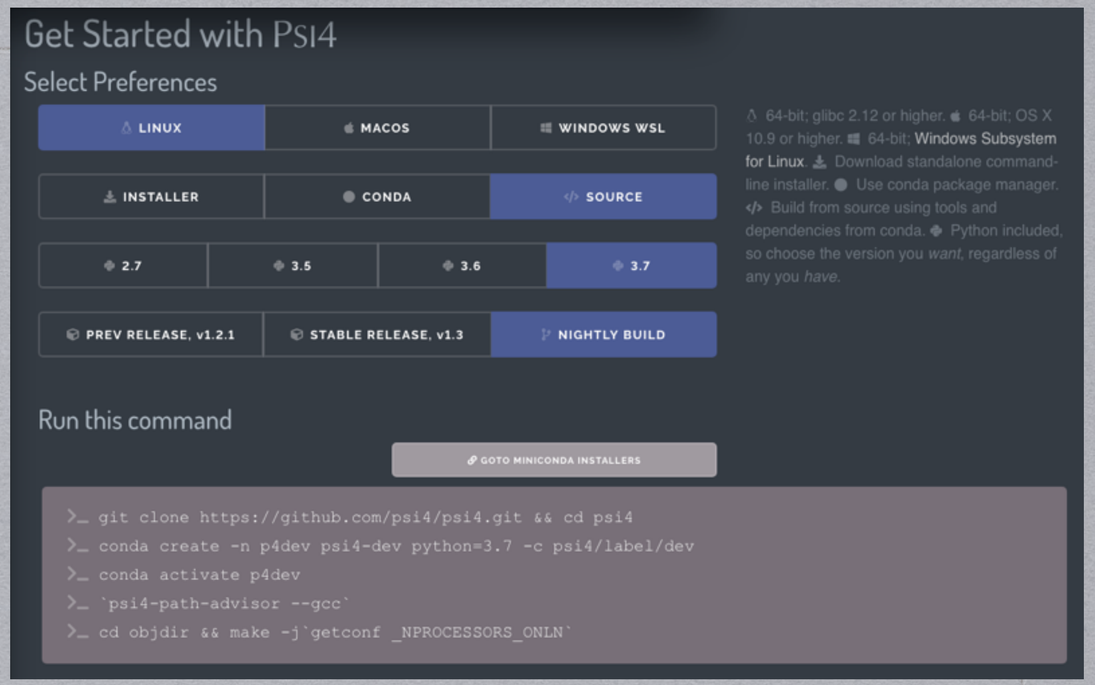
PY-WARY USERS: CONDA INSTALLER

PSI4 · dependencies · add-ons

PSI4CONDA 57K

CONDA FOR USERS & DEVELOPERS





CORE DEVELOPERS: CONDA-ENABLED SUPERBUILD

deps · dev tools · add-ons · cmake hints

PY-FRIENDLY USERS: CONDA PACKAGE

PSI4 · dependencies · add-ons

50K

PY-WARY USERS: CONDA INSTALLER

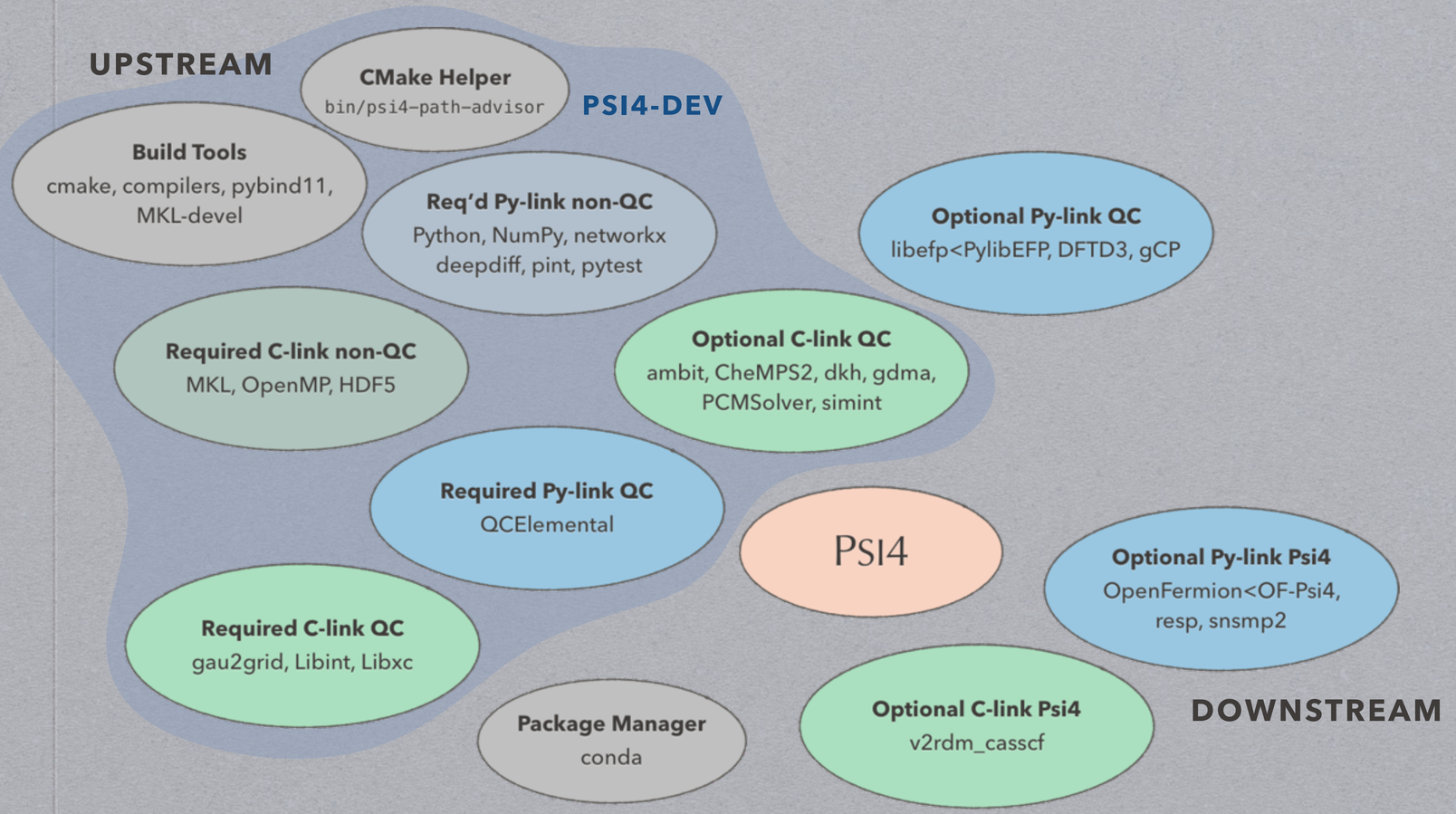
PSI4 · dependencies · add-ons

PSI4CONDA

57K

CONDA FOR USERS & DEVELOPERS

ANACONDA CLOUD



PERIPHERAL DEVEL'S: CONDA-FACILITATED PLUGINS
PSI4 · dev tools · cmake hints

CORE DEVELOPERS: CONDA-ENABLED SUPERBUILD
deps · dev tools · add-ons · cmake hints

PY-FRIENDLY USERS: CONDA PACKAGE 50K
PSI4 · dependencies · add-ons

PY-WARY USERS: CONDA INSTALLER PSI4CONDA 57K
PSI4 · dependencies · add-ons

PRODUCTION-QUALITY BINARIES → STANDARDS

DEVS CAN HAVE SHINY THINGS

PRODUCTION DISTRIBUTIONS

- **LAPACK** dynamically link runtime-multiarch MKL direct from Intel.
- **MULTIARCH** libs optimized for both modern & legacy arch via `icpc` flags.
- **STANDARDS** easy to distribute with gcc7.3, so devs can use c++14.
- **COMPATIBILITY** conda compilers have sysroots with old glibc so useable with slow-moving Linux OS. Binary-compatible w/ defaults & conda-forge channels.

STANDARDS

- **SHINY THINGS** advance standards liberally.
  - balance with user ease
  - balance with distribution ease
  - balance with not imposing version freeze on other projects
- **PYTHON** 3.6+ after mid-2018.
- **C++** 14 by Dec 2018.
- **CONDA COMPILERS**
  - **LINUX** GCC 7.3 for C, C++, Fortran. Psi uses ICPC 2018 atop GCC for multiarch opt.
  - **MAC** Clang 4.0.1 for C, C++; GCC Fortran.
  - **WIN** MSVC for C, C++; IFORT for Fortran.

DECLARATIVE INTERFACE

ALLOWS TRANSPARENT REFACTORING BETWEEN MODULES, LANGS, REPOS

- **ENERGY()**, **`gradient()`**, **`optimize()`**, **`hessian()`**, **`frequency()`** are the five "user-facing" functions through which 99+% of QC is run in Psi, so easy to guess command. Minimal entry points
- **BEST-PRACTICE OPTIONS** for basis sets, convergence, implementation, algorithm are added at driver layer.
- **PY/C++** interface layer below the user layer makes it simple to shift methods btwn languages without disturbing user.
  - **PY → C++** when a reference implementation is optimized in compiled language.
  - **C++ → PY** when a legacy code is refactored so that logic in Python and intensive parts in compiled.
  - **E.G. MOLECULE** parsing
    - C++ with Boost regex on str (v1.0)
    - C++ with C++11 regex on str (v1.1)
    - Py dict initializing class (v1.2)
    - Py dict from external module initializing class (v1.3)
  - **E.G. B3LYP**
    - in-house fctl + lib3index DF + Wfn::KS (v1.1)
    - Libxc fctl + lib3index DF + Wfn::KS
    - Libxc fctl + lib3index DF + Wfn::SCF
    - Libxc fctl + DF_Helper DF + Wfn::SCF (v1.2)

```
psi4.energy('mp2')
```

```
psi4.energy('mp2/cc-pvtz')
```

```
psi4.energy('mp2/cc-pv[dt]z')
```

```
psi4.gradient('mp2/cc-pvtz', bsse_type='cp')
```

```
psi4.optimize('mp2/cc-pv[dt]', bsse_type='uncp', dertype=0)
```

DECLARATIVE INTERFACE

ALLOWS TRANSPARENT REFACTORING BETWEEN MODULES, LANGS, REPOS

- **ENERGY()**, **`gradient()`**, **`optimize()`**, **`hessian()`**, **`frequency()`** are the five "user-facing" functions through which 99+% of QC is run in Psi, so easy to guess command. Minimal entry points
- **BEST-PRACTICE OPTIONS** for basis sets, convergence, implementation, algorithm are added at driver layer.
- **PY/C++** interface layer below the user layer makes it simple to shift methods btwn languages without disturbing user.
  - **PY → C++** when a reference implementation is optimized in compiled language.
  - **C++ → PY** when a legacy code is refactored so that logic in Python and intensive parts in compiled.
  - **E.G. MOLECULE** parsing
    - C++ with Boost regex on str (v1.0)
    - C++ with C++11 regex on str (v1.1)
    - Py dict initializing class (v1.2)
    - Py dict from external module initializing class (v1.3)
  - **E.G. B3LYP**
    - in-house fctl + lib3index DF + Wfn::KS (v1.1)
    - Libxc fctl + lib3index DF + Wfn::KS
    - Libxc fctl + lib3index DF + Wfn::SCF
    - Libxc fctl + DF_Helper DF + Wfn::SCF (v1.2)

REVOLUTIONS

wavefunction passing - localizing, in-memory, givinguser access to calc innards – 2016

molecule passing - globals avoidance – 2016

recursivedriver, minimal entry points – 2016

CMakeRewrite – 2016

dependency ejection (build-wise) – 2016

KillTheBoost – 2016

HistoryRewrite – 2016

pysidescf – 2018

molparse – 2018

qcvar localization - globals avoidance, definition coherency – 2018–present

distributeddriver – 2018–present

dependency ejection (interface-wise to QCA stack) – 2019–present

theBeheading – upcoming

options passing - globals avoidance – upcoming

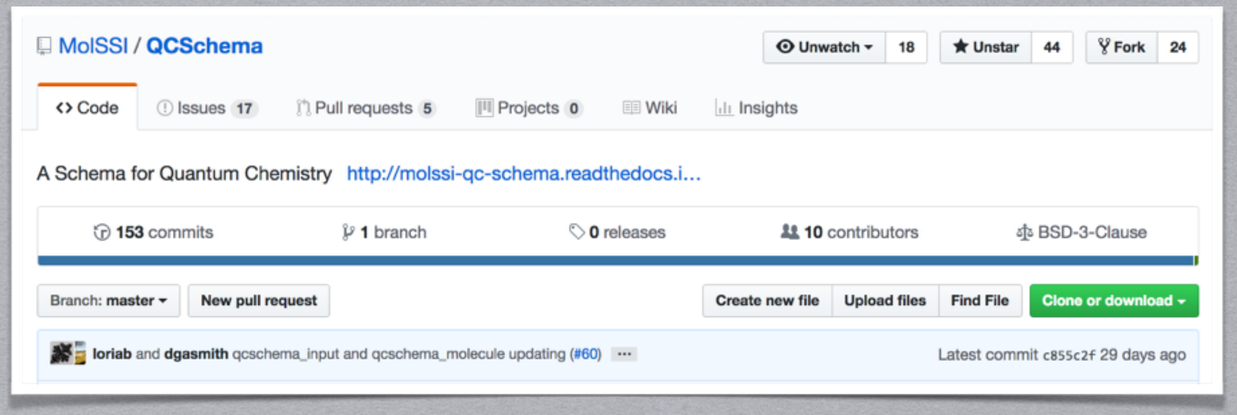
QCSchema

FULL-FLEDGED SINGLE JOB SPEC

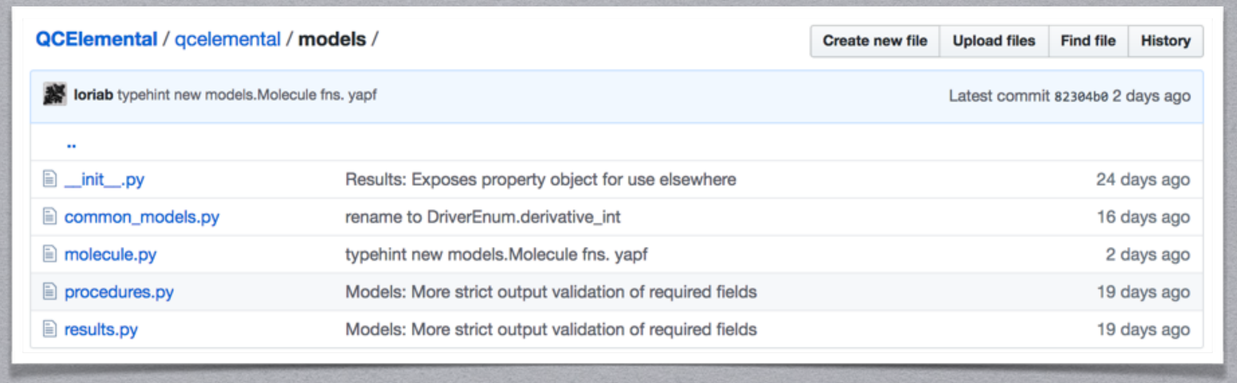
QCJOB DICT

- **CHARACTERISTICS**
  - describes single QC step in unified language
  - DICT, non-serializable, conforms to schema
  - filled by translator functions
- **CONTENTS**
  - **DIRECTIONS** QCprog, method, basis, deriv level
  - **OPTIONS** multilevel, history, Py-format values
  - **RESOURCES** exe loc, scratch, mem, threads
  - **OUTPUTS** stdout, qcvars, interpret-time errors

DISCUSSION HERE

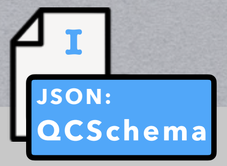


VALIDATING PYTHON OBJECTS HERE



QCSchema

FULL-FLEDGED SINGLE JOB SPEC



QCJOB DICT

- **CHARACTERISTICS**
  - describes single QC step in unified language
  - DICT, non-serializable, conforms to schema
  - filled by translator functions
- **CONTENTS**
  - **DIRECTIONS** QCprog, method, basis, deriv level
  - **OPTIONS** multilevel, history, Py-format values
  - **RESOURCES** exe loc, scratch, mem, threads
  - **OUTPUTS** stdout, qcvars, interpret-time errors

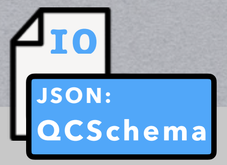
```
{
   'driver': 'gradient',
   'model': {'method': 'ccsd(t)',
             'basis': 'cc-pvdz'},
   'molecule': {'atomic_numbers': [8, 1, 1],
                ...
                'geometry': [0.0, 0.0, -0.065638538099178
                'fix_symmetry': 'Cs'},
   'keywords': {'freeze_core': True,
                ...
                'r_convergence': 8},
   'provenance': None,
   'success': None
}
```

QCSchema

FULL-FLEDGED SINGLE JOB SPEC

QCJOB DICT

- **CHARACTERISTICS**
  - describes single QC step in unified language
  - DICT, non-serializable, conforms to schema
  - filled by translator functions
- **CONTENTS**
  - **DIRECTIONS** QCprog, method, basis, deriv level
  - **OPTIONS** multilevel, history, Py-format values
  - **RESOURCES** exe loc, scratch, mem, threads
  - **OUTPUTS** stdout, qcvars, interpret-time errors



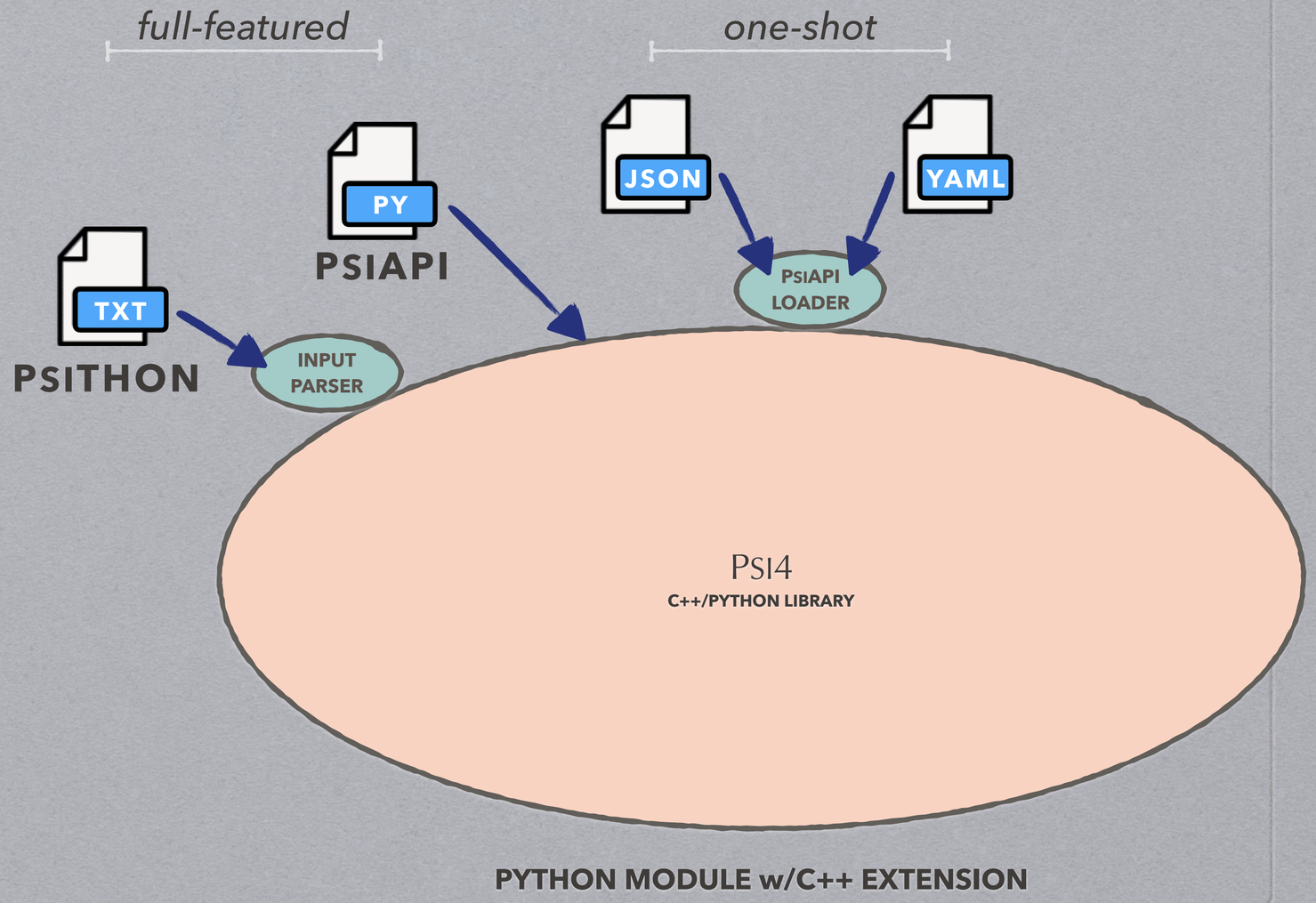

```
{
  'driver': DITTO
  'model': DITTO
  'molecule': DITTO
  'keywords': DITTO
  'provenance': {'by': 'Psi4', 'func': 'run_json', 'version': '1.3'},
  'raw_output': '    * CFOUR Coupled-Cluster techniques for Computational Chemistry *\n'
                '      CALCLEVEL           ICLLVL          CCSD(T)    [ 22]   ***   \n'
                '      CC_PROGRAM          ICCPRO          ECC        [  1]   ***   \n'
                ' A miracle come to pass. The CC iterations have converged.\n'
                '  CCSD(T) energy                      -76.320175532159\n'
                '  Cfour scratch file GRD has been read\n'
                '    3       0.0000000000\n'
                '       8.0000000000       0.0000000000       0.0000000000       0.0006534633\n'
                '       1.0000000000       0.0000000000      -0.0003521950      -0.0003267317\n'
                '       1.0000000000       0.0000000000       0.0003521950      -0.0003267317\n'
  'extras': {'qcvars': {
    '(T) CORRECTION ENERGY': -0.0070965085890132,
    'CCSD CORRELATION ENERGY': -0.2503305491879538,
    'CCSD(T) TOTAL ENERGY': -76.320175532159,
    'CURRENT GRADIENT': [[ 0.        ,  0.        ,  0.00065346],
      [ 0.        , -0.0003522 , -0.00032673],
      [ 0.        ,  0.0003522 , -0.00032673]]}},
  'success': True
}
```

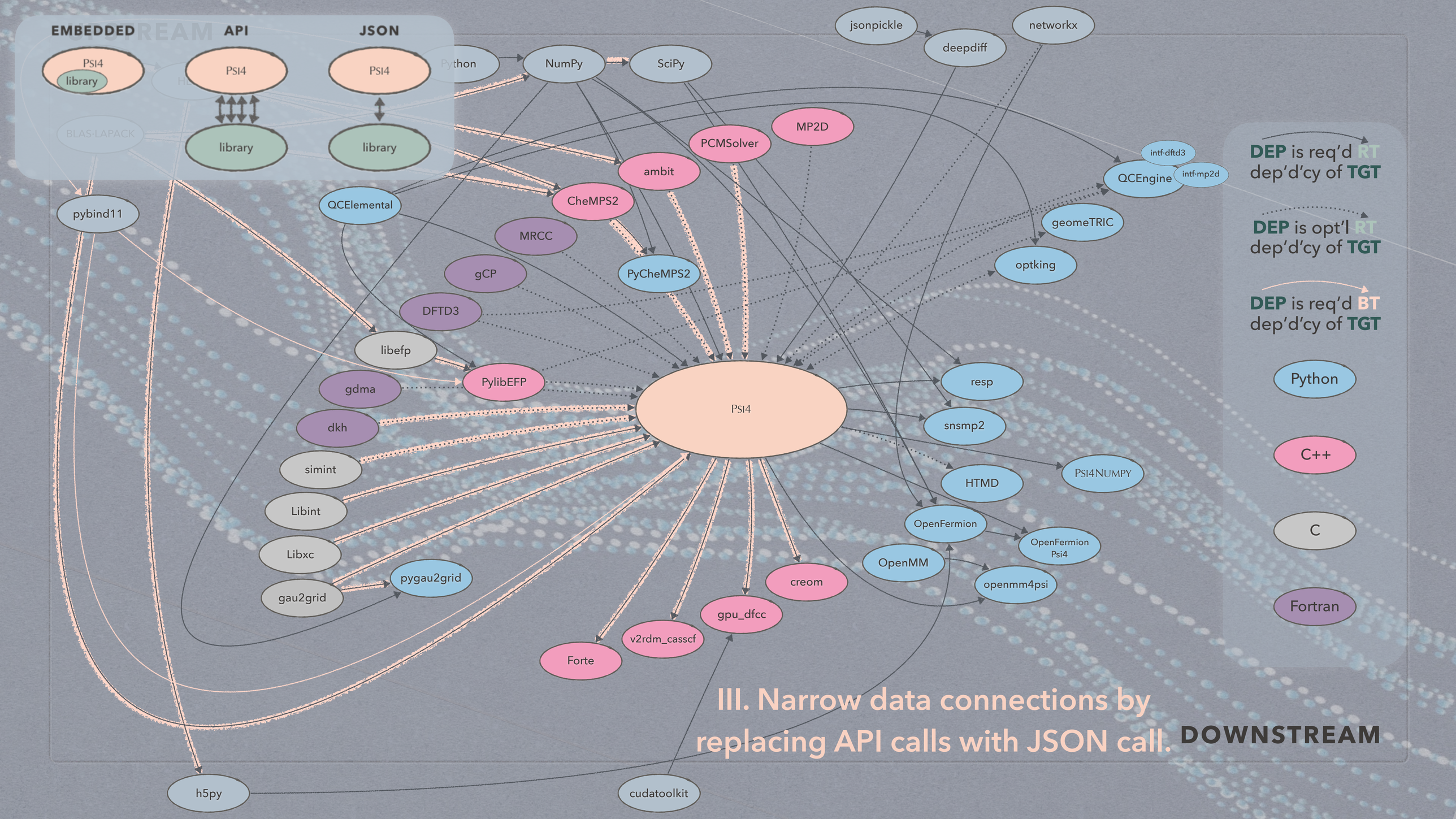
QCSchema

FULL-FLEDGED SINGLE JOB SPEC

QCJOB DICT

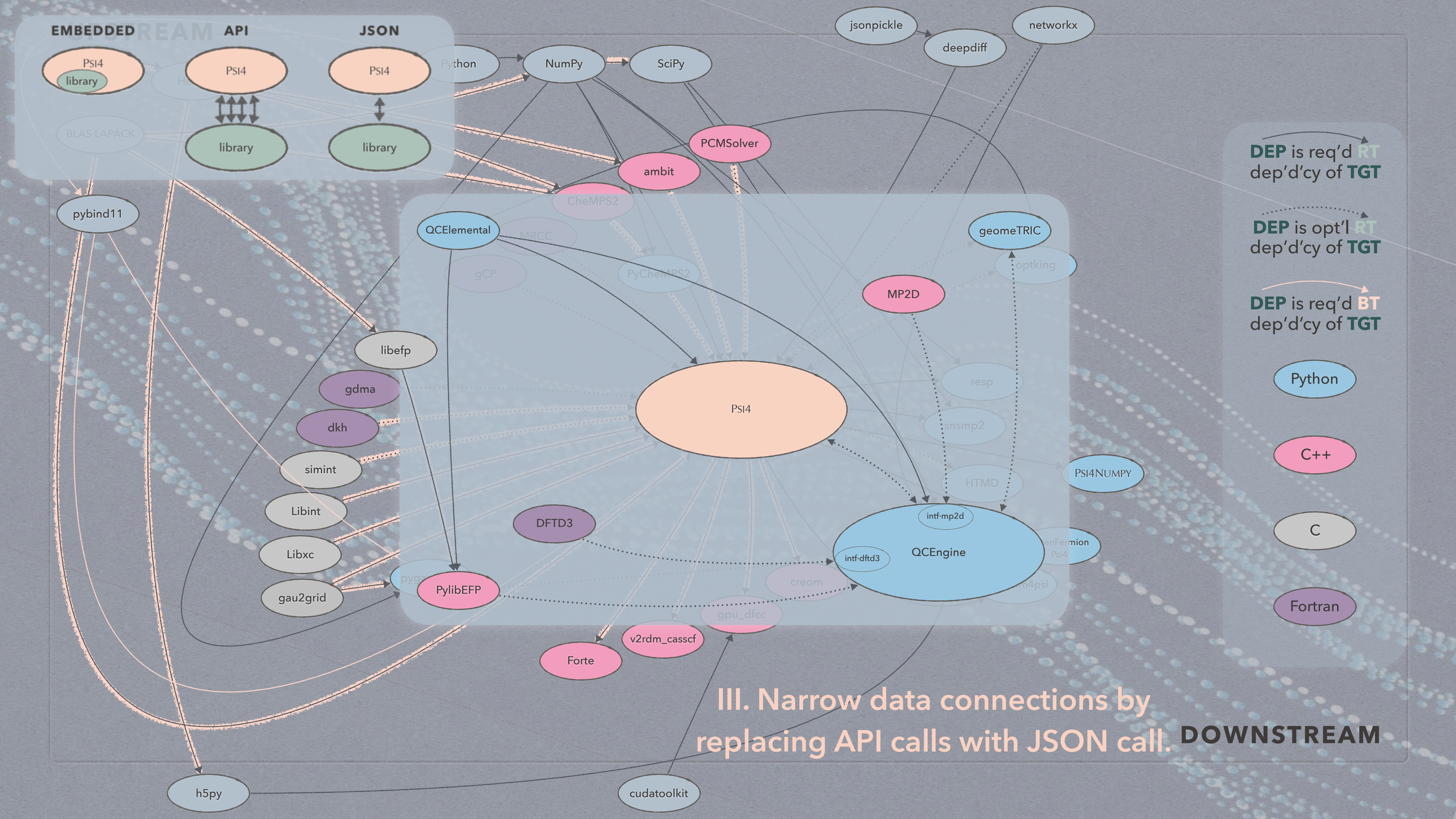
- **CHARACTERISTICS**
  - describes single QC step in unified language
  - DICT, non-serializable, conforms to schema
  - filled by translator functions
- **CONTENTS**
  - **DIRECTIONS** QCprog, method, basis, deriv level
  - **OPTIONS** multilevel, history, Py-format values
  - **RESOURCES** exe loc, scratch, mem, threads
  - **OUTPUTS** stdout, qcvars, interpret-time errors





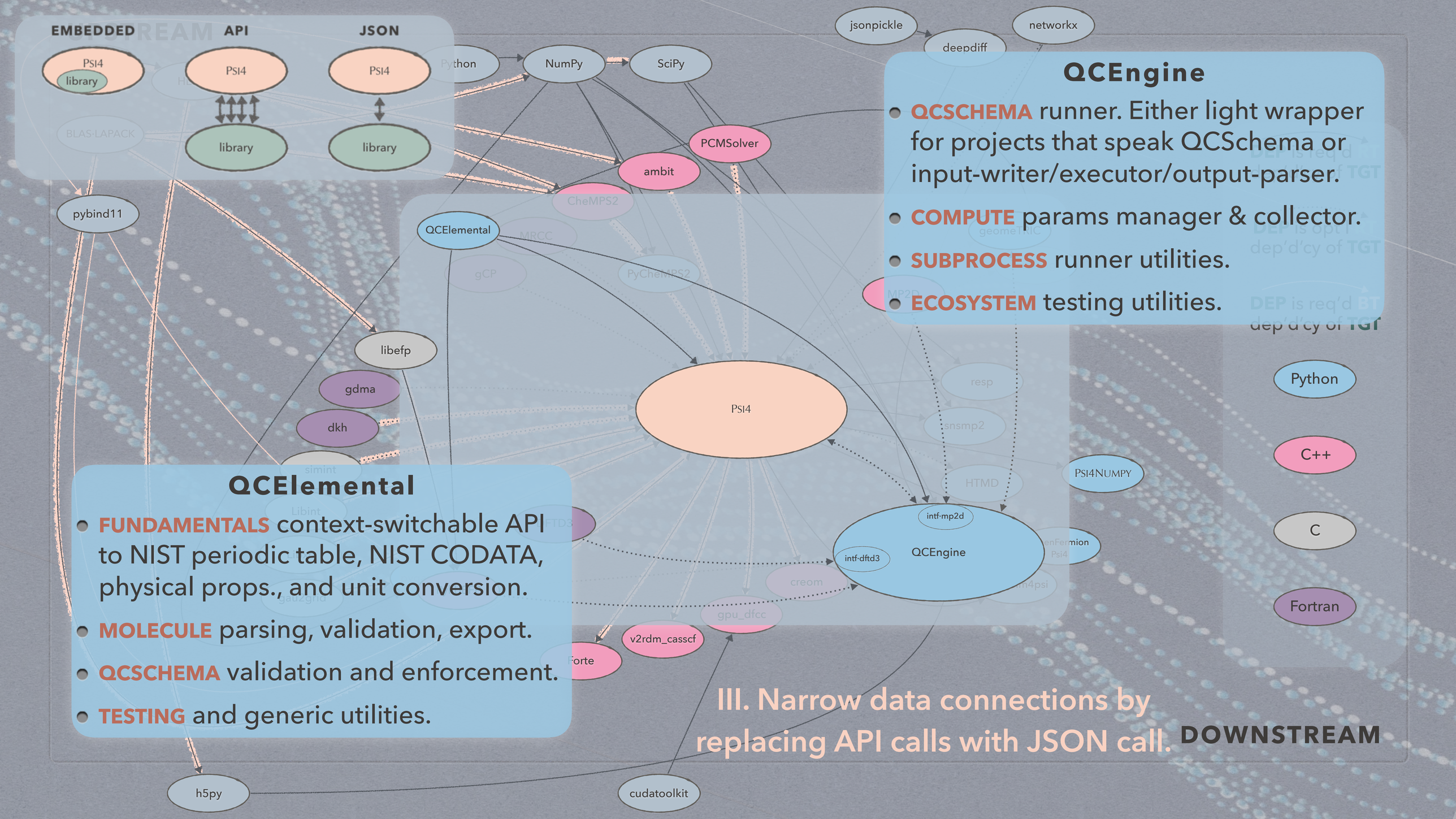
III. Narrow data connections by replacing API calls with JSON call.

DOWNSTREAM



III. Narrow data connections by replacing API calls with JSON call.

DOWNSTREAM



QCEngine

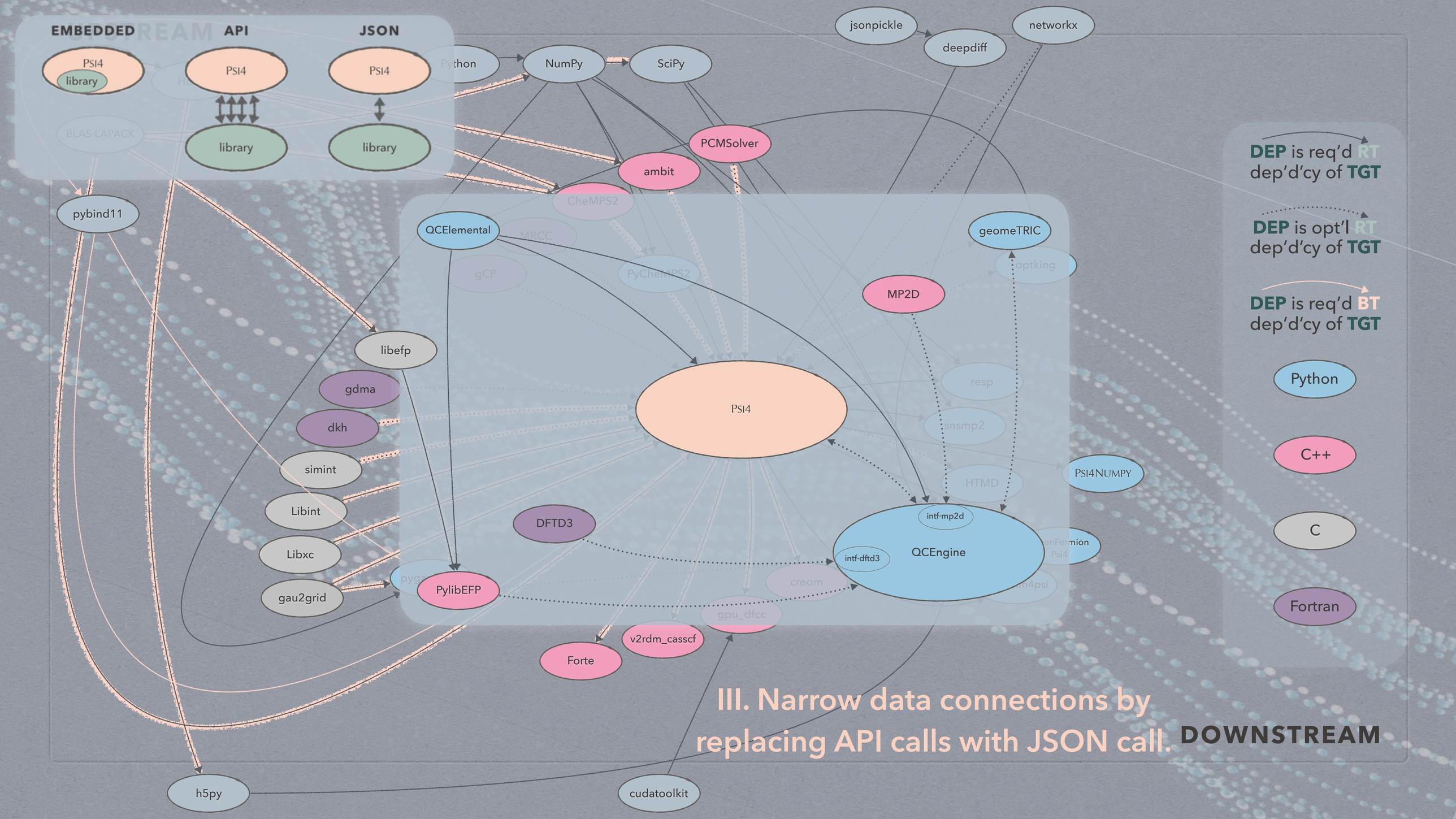
- **QCSCHEMA** runner. Either light wrapper for projects that speak QCSchema or input-writer/executor/output-parser.
- **COMPUTE** params manager & collector.
- **SUBPROCESS** runner utilities.
- **ECOSYSTEM** testing utilities.

QCElemental

- **FUNDAMENTALS** context-switchable API to NIST periodic table, NIST CODATA, physical props., and unit conversion.
- **MOLECULE** parsing, validation, export.
- **QCSCHEMA** validation and enforcement.
- **TESTING** and generic utilities.

III. Narrow data connections by replacing API calls with JSON call.

DOWNSTREAM



III. Narrow data connections by replacing API calls with JSON call.

DOWNSTREAM

DISTRIBUTED DRIVER

```
def energy (mtd):
```

```
def gradient (mtd):
```

```
def hessian (mtd):
```

DISTRIBUTED DRIVER

```
def energy (mtd):
```

```
def gradient (mtd):
```

```
def hessian (mtd):
```

```
class SingleResult ():
```

PLAN **molecule** & **method** & **func** unchanged. `return json`

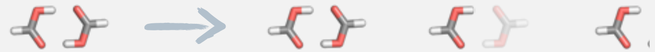
ASM Return analytic energy, gradient, or Hessian.

DISTRIBUTED DRIVER

`def energy (mtd):` `def gradient (mtd):` `def hessian (mtd):`

`class NBodyComputer ():`

PLAN Separate **`molecule`** into subsystems. CP, noCP, VMFC basis. **`method`** & **`func`** unchanged.



`for frag in fragments: return json`

ASM Assemble n-body & interaction results from fragments.

`class SingleResult ():`

PLAN **`molecule`** & **`method`** & **`func`** unchanged. `return json`

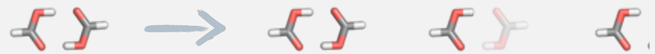
ASM Return analytic energy, gradient, or Hessian.

DISTRIBUTED DRIVER

`def energy (mtd):` `def gradient (mtd):` `def hessian (mtd):`

`class NBodyComputer ():`

PLAN Separate **`molecule`** into subsystems. CP, noCP, VMFC basis. **`method`** & **`func`** unchanged.



`for frag in fragments: return json`

ASM Assemble n-body & interaction results from fragments.

`class CBSComputer ():`

PLAN Separate **`method`** into method, basis, & extrapolations. **`molecule`** & **`func`** unchanged.

`'mp2/cc-pv[tq]z'` → **`MP2 TOTAL ENERGY/cc-pVTZ`** **`MP2 TOTAL ENERGY/cc-pVQZ`**

`for mc in modelchems: return json`

ASM Assemble extrapolations & total results from modelchems.

`class SingleResult ():`

PLAN **`molecule`** & **`method`** & **`func`** unchanged. `return json`

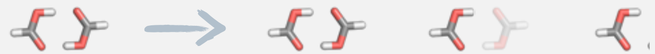
ASM Return analytic energy, gradient, or Hessian.

DISTRIBUTED DRIVER

`def energy (mtd):` `def gradient (mtd):` `def hessian (mtd):`

`class NBodyComputer ():`

PLAN Separate **molecule** into subsystems. CP, noCP, VMFC basis.
method & **func** unchanged.



`for frag in fragments: return json`

ASM Assemble n-body & interaction results from fragments.

`class CBSComputer ():`

PLAN Separate **method** into method, basis, & extrapolations.
molecule & **func** unchanged.

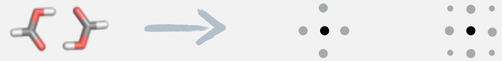
`'mp2/cc-pv[tq]z'` → `MP2 TOTAL ENERGY/cc-pVTZ`
`MP2 TOTAL ENERGY/cc-pVQZ`

`for mc in modelchems: return json`

ASM Assemble extrapolations & total results from modelchems.

`class FinDifComputer ():`

PLAN Displace **molecule** according to stencil.
Reference **molecule** & **func** unchanged.



`for disp in displacements: return json`

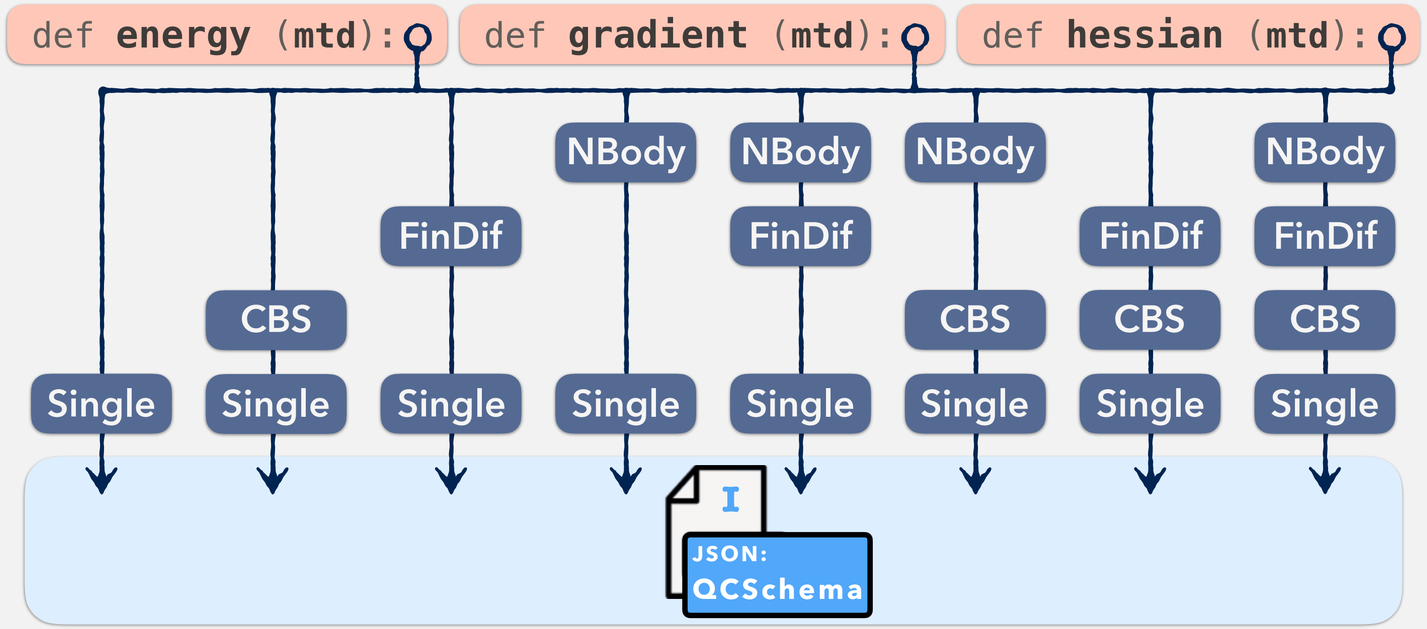
ASM Assemble derivative results from displacements.

`class SingleResult ():`

PLAN **molecule** & **method** & **func** unchanged. `return json`

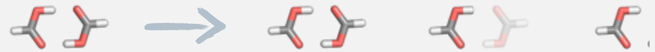
ASM Return analytic energy, gradient, or Hessian.

DISTRIBUTED DRIVER



class NBodyComputer ():

PLAN Separate **molecule** into subsystems. CP, noCP, VMFC basis. **method** & **func** unchanged.



```
for frag in fragments: return json
```

ASM Assemble n-body & interaction results from fragments.

class CBSComputer ():

PLAN Separate **method** into method, basis, & extrapolations. **molecule** & **func** unchanged.

'mp2/cc-pv[tq]z' → **MP2 TOTAL ENERGY/cc-pVTZ**
MP2 TOTAL ENERGY/cc-pVQZ

```
for mc in modelchems: return json
```

ASM Assemble extrapolations & total results from modelchems.

class FinDifComputer ():

PLAN Displace **molecule** according to stencil. Reference **molecule** & **func** unchanged.

```
for disp in displacements: return json
```

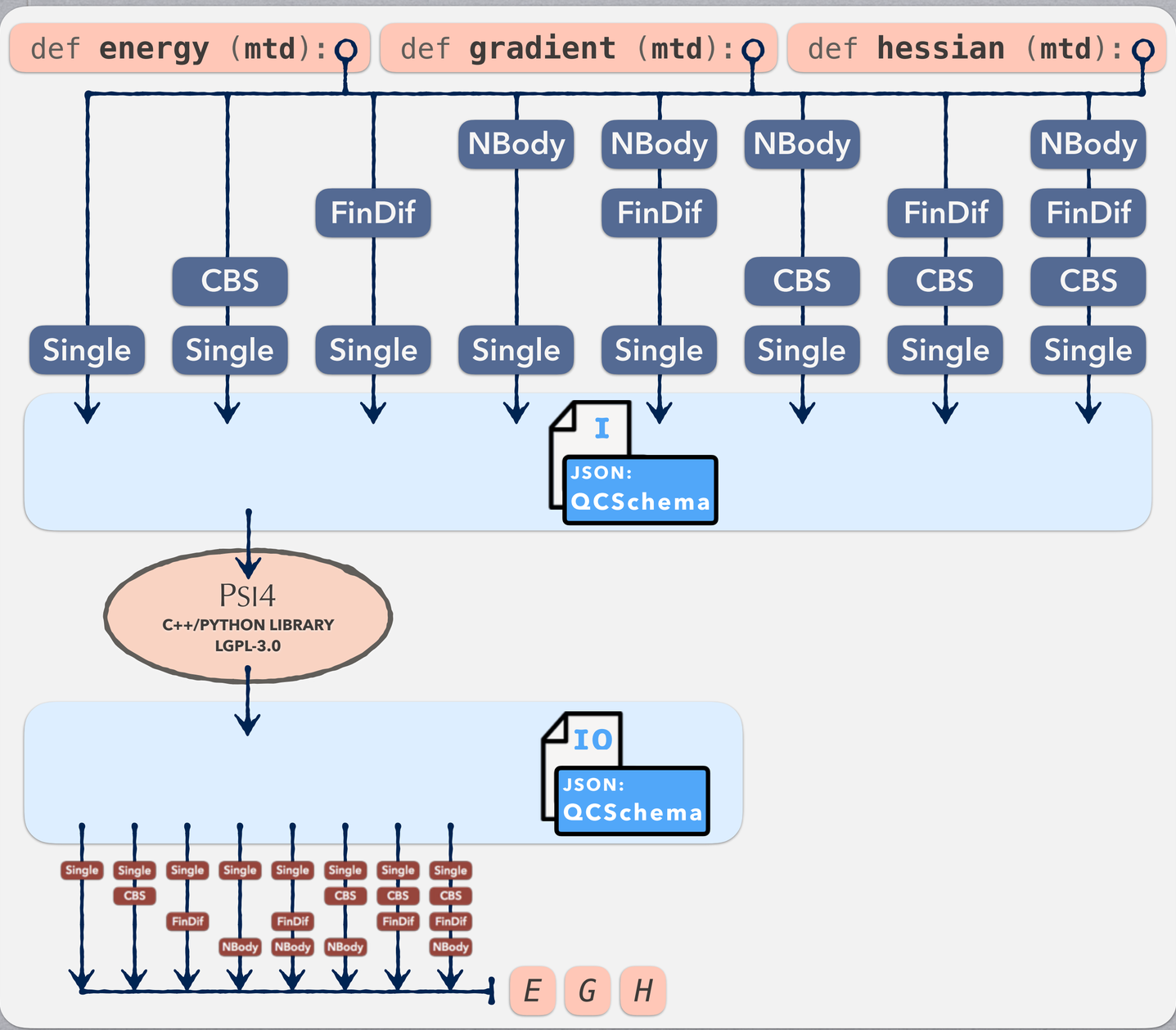
ASM Assemble derivative results from displacements.

class SingleResult ():

PLAN **molecule** & **method** & **func** unchanged. `return json`

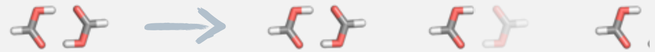
ASM Return analytic energy, gradient, or Hessian.

DISTRIBUTED DRIVER



class NBodyComputer ():

PLAN Separate **molecule** into subsystems. CP, noCP, VMFC basis.
method & **func** unchanged.



```
for frag in fragments: return json
```

ASM Assemble n-body & interaction results from fragments.

class CBSComputer ():

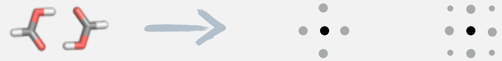
PLAN Separate **method** into method, basis, & extrapolations.
molecule & **func** unchanged.

```
'mp2/cc-pv[tq]z'  →  MP2 TOTAL ENERGY/cc-pVTZ
                     MP2 TOTAL ENERGY/cc-pVQZ
for mc in modelchems: return json
```

ASM Assemble extrapolations & total results from modelchems.

class FinDifComputer ():

PLAN Displace **molecule** according to stencil.
Reference **molecule** & **func** unchanged.



```
for disp in displacements: return json
```

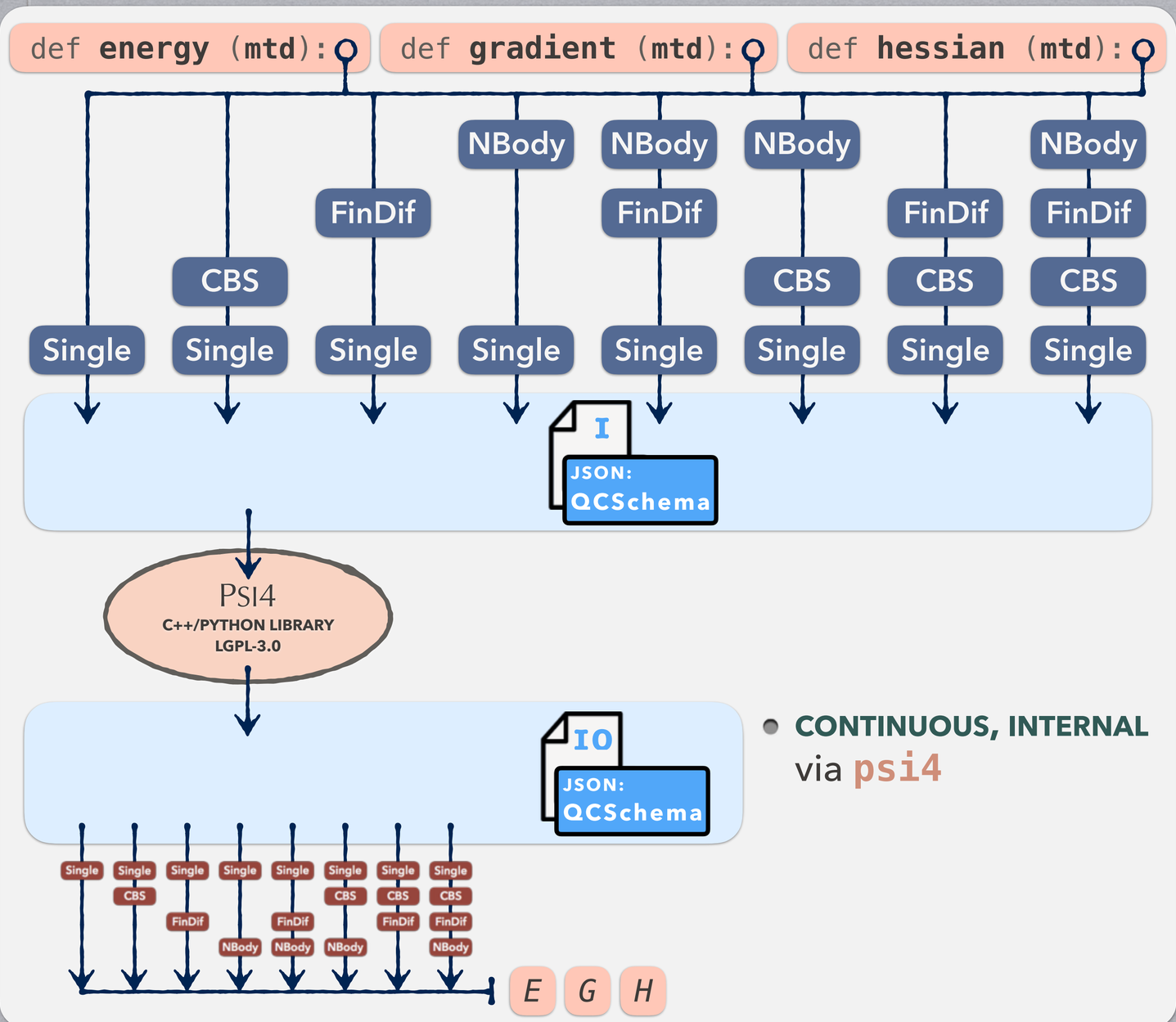
ASM Assemble derivative results from displacements.

class SingleResult ():

PLAN **molecule** & **method** & **func** unchanged. `return json`

ASM Return analytic energy, gradient, or Hessian.

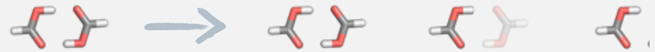
DISTRIBUTED DRIVER



- **CONTINUOUS, INTERNAL**
  via `psi4`

`class NBodyComputer ():`

PLAN Separate **`molecule`** into subsystems. CP, noCP, VMFC basis.
`method` & **`func`** unchanged.



```
for frag in fragments: return json
```

ASM Assemble n-body & interaction results from fragments.

`class CBSComputer ():`

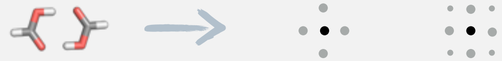
PLAN Separate **`method`** into method, basis, & extrapolations.
`molecule` & **`func`** unchanged.

```
'mp2/cc-pv[tq]z'  →  MP2 TOTAL ENERGY/cc-pVTZ
                     MP2 TOTAL ENERGY/cc-pVQZ
for mc in modelchems: return json
```

ASM Assemble extrapolations & total results from modelchems.

`class FinDifComputer ():`

PLAN Displace **`molecule`** according to stencil.
Reference **`molecule`** & **`func`** unchanged.



```
for disp in displacements: return json
```

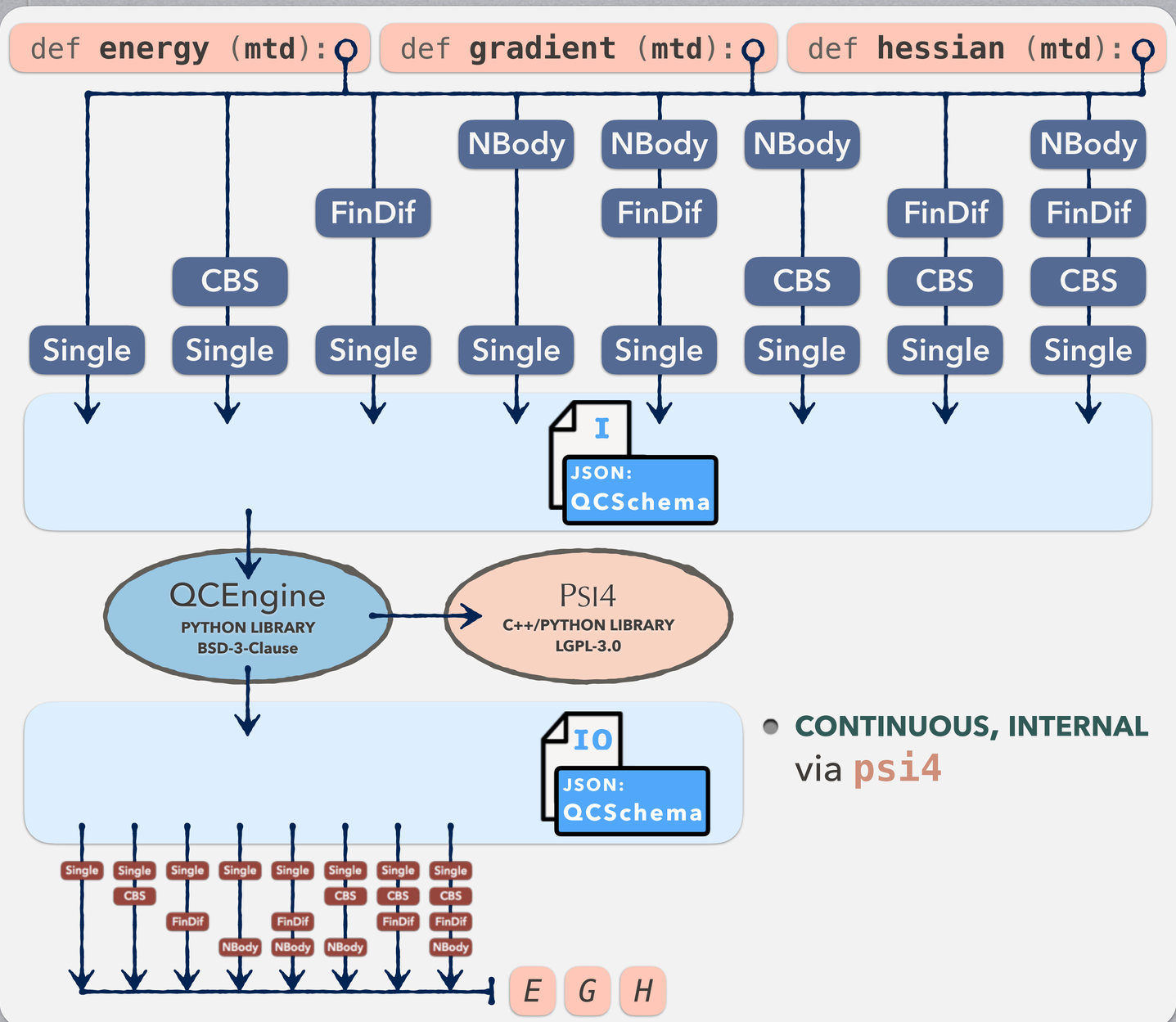
ASM Assemble derivative results from displacements.

`class SingleResult ():`

PLAN **`molecule`** & **`method`** & **`func`** unchanged. `return json`

ASM Return analytic energy, gradient, or Hessian.

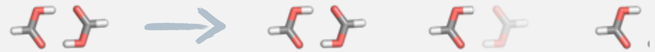
DISTRIBUTED DRIVER



- **CONTINUOUS, INTERNAL** via `psi4`

`class NBodyComputer ():`

PLAN Separate **`molecule`** into subsystems. CP, noCP, VMFC basis. **`method`** & **`func`** unchanged.



```
for frag in fragments: return json
```

ASM Assemble n-body & interaction results from fragments.

`class CBSComputer ():`

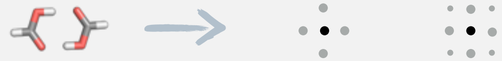
PLAN Separate **`method`** into method, basis, & extrapolations. **`molecule`** & **`func`** unchanged.

```
'mp2/cc-pv[tq]z'  →  MP2 TOTAL ENERGY/cc-pVTZ
                     MP2 TOTAL ENERGY/cc-pVQZ
for mc in modelchems: return json
```

ASM Assemble extrapolations & total results from modelchems.

`class FinDifComputer ():`

PLAN Displace **`molecule`** according to stencil. Reference **`molecule`** & **`func`** unchanged.



```
for disp in displacements: return json
```

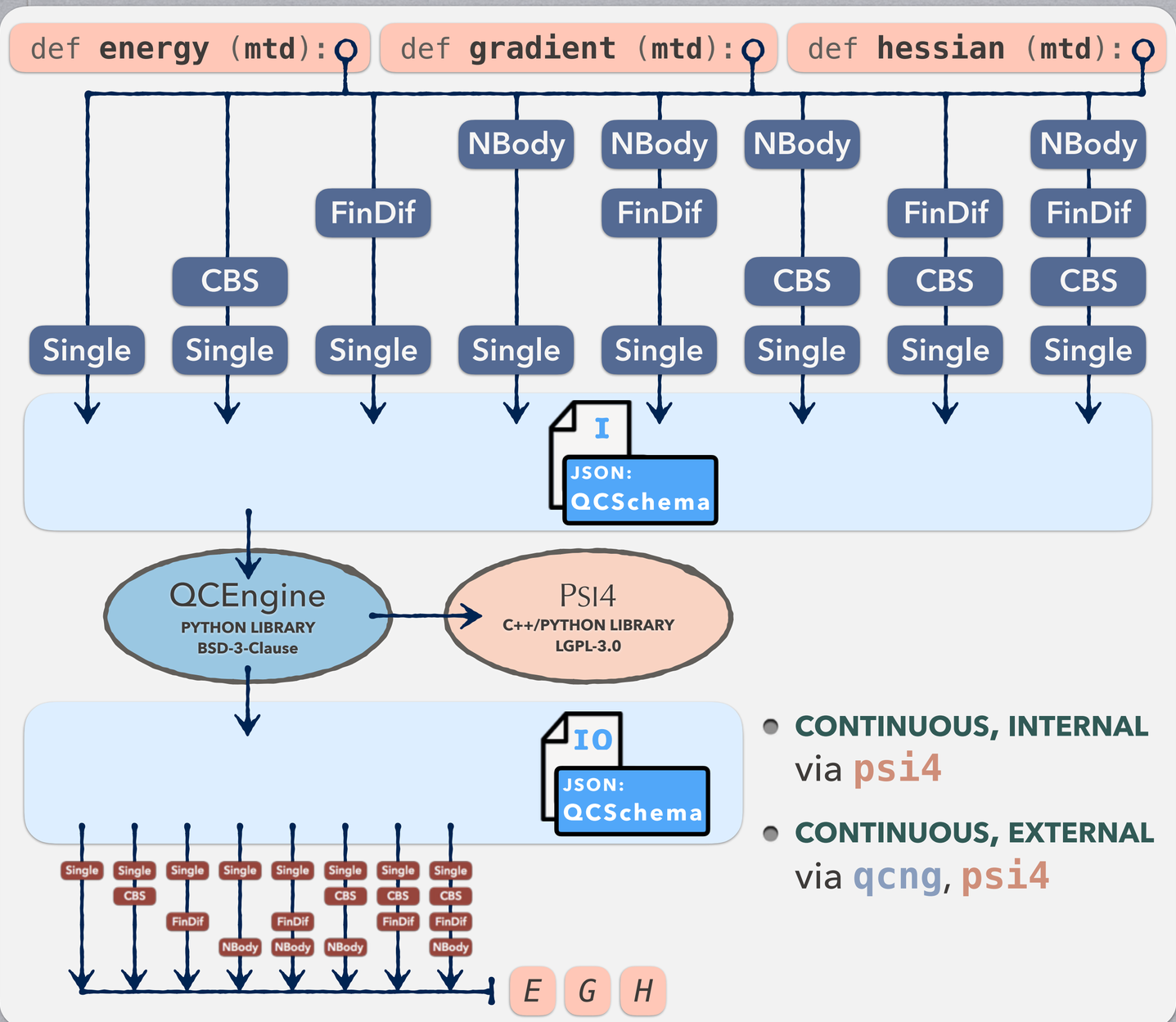
ASM Assemble derivative results from displacements.

`class SingleResult ():`

PLAN **`molecule`** & **`method`** & **`func`** unchanged. `return json`

ASM Return analytic energy, gradient, or Hessian.

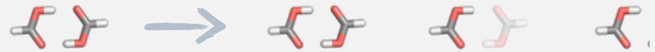
DISTRIBUTED DRIVER



- **CONTINUOUS, INTERNAL** via `psi4`
- **CONTINUOUS, EXTERNAL** via `qcng`, `psi4`

`class NBodyComputer ():`

PLAN Separate **`molecule`** into subsystems. CP, noCP, VMFC basis. **`method`** & **`func`** unchanged.



`for frag in fragments: return json`

ASM Assemble n-body & interaction results from fragments.

`class CBSComputer ():`

PLAN Separate **`method`** into method, basis, & extrapolations. **`molecule`** & **`func`** unchanged.

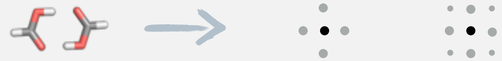
`'mp2/cc-pv[tq]z'` → `MP2 TOTAL ENERGY/cc-pVTZ`, `MP2 TOTAL ENERGY/cc-pVQZ`

`for mc in modelchems: return json`

ASM Assemble extrapolations & total results from modelchems.

`class FinDifComputer ():`

PLAN Displace **`molecule`** according to stencil. Reference **`molecule`** & **`func`** unchanged.



`for disp in displacements: return json`

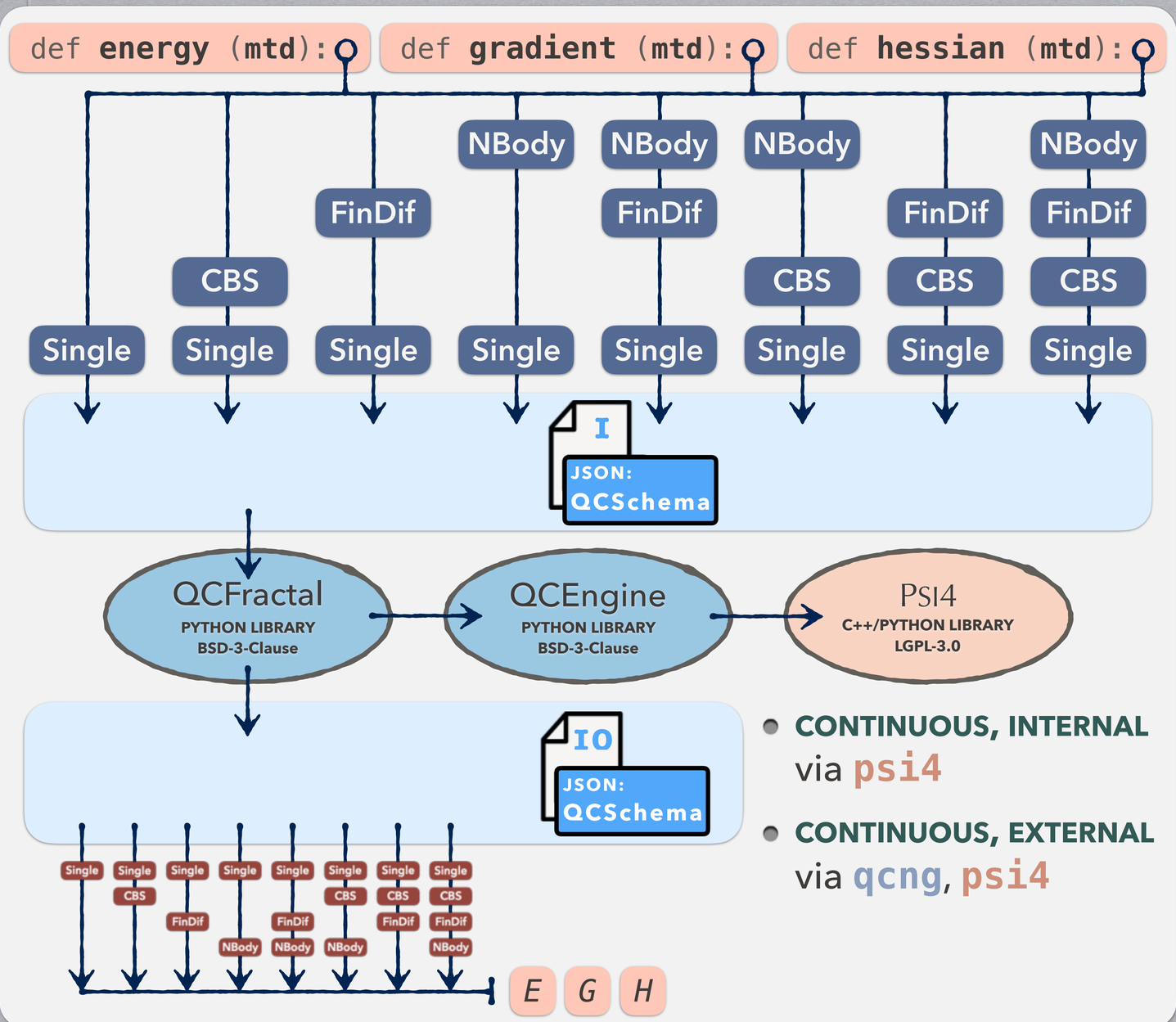
ASM Assemble derivative results from displacements.

`class SingleResult ():`

PLAN **`molecule`** & **`method`** & **`func`** unchanged. `return json`

ASM Return analytic energy, gradient, or Hessian.

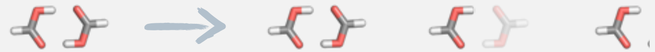
DISTRIBUTED DRIVER



- **CONTINUOUS, INTERNAL** via `psi4`
- **CONTINUOUS, EXTERNAL** via `qcng`, `psi4`

`class NBodyComputer ():`

PLAN Separate **`molecule`** into subsystems. CP, noCP, VMFC basis. **`method`** & **`func`** unchanged.



`for frag in fragments: return json`

ASM Assemble n-body & interaction results from fragments.

`class CBSComputer ():`

PLAN Separate **`method`** into method, basis, & extrapolations. **`molecule`** & **`func`** unchanged.

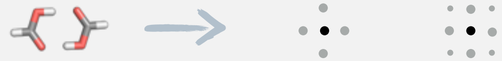
`'mp2/cc-pv[tq]z'` → `MP2 TOTAL ENERGY/cc-pVTZ` `MP2 TOTAL ENERGY/cc-pVQZ`

`for mc in modelchems: return json`

ASM Assemble extrapolations & total results from modelchems.

`class FinDifComputer ():`

PLAN Displace **`molecule`** according to stencil. Reference **`molecule`** & **`func`** unchanged.



`for disp in displacements: return json`

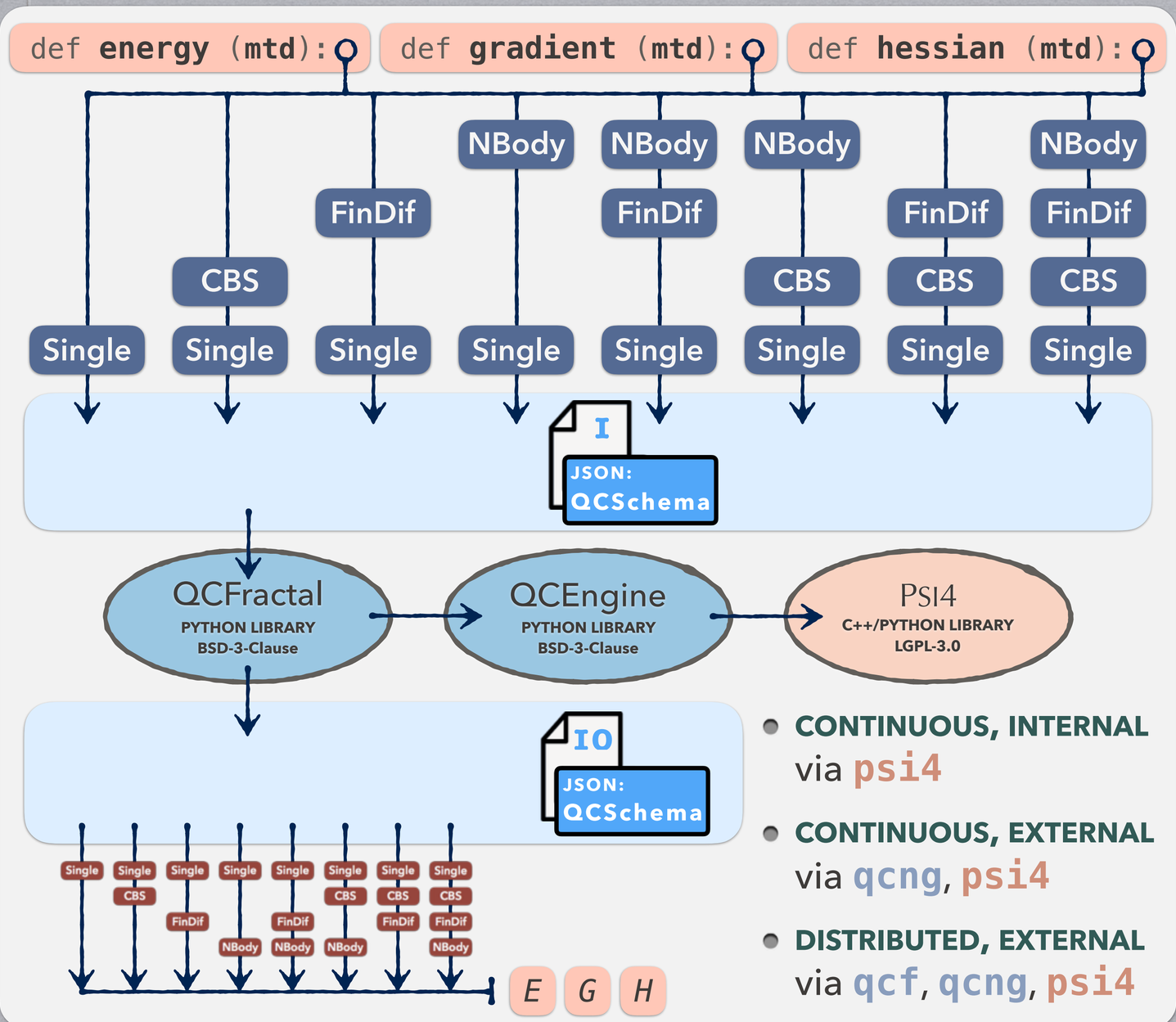
ASM Assemble derivative results from displacements.

`class SingleResult ():`

PLAN **`molecule`** & **`method`** & **`func`** unchanged. `return json`

ASM Return analytic energy, gradient, or Hessian.

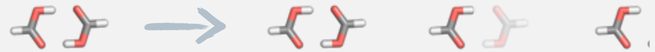
DISTRIBUTED DRIVER



- **CONTINUOUS, INTERNAL** via `psi4`
- **CONTINUOUS, EXTERNAL** via `qcng`, `psi4`
- **DISTRIBUTED, EXTERNAL** via `qcf`, `qcng`, `psi4`

`class NBodyComputer ():`

PLAN Separate **`molecule`** into subsystems. CP, noCP, VMFC basis. **`method`** & **`func`** unchanged.



```
for frag in fragments: return json
```

ASM Assemble n-body & interaction results from fragments.

`class CBSComputer ():`

PLAN Separate **`method`** into method, basis, & extrapolations. **`molecule`** & **`func`** unchanged.

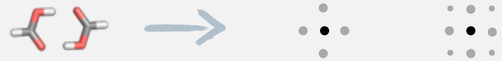
`'mp2/cc-pv[tq]z'` → `MP2 TOTAL ENERGY/cc-pVTZ`, `MP2 TOTAL ENERGY/cc-pVQZ`

```
for mc in modelchems: return json
```

ASM Assemble extrapolations & total results from modelchems.

`class FinDifComputer ():`

PLAN Displace **`molecule`** according to stencil. Reference **`molecule`** & **`func`** unchanged.



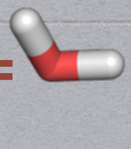
```
for disp in displacements: return json
```

ASM Assemble derivative results from displacements.

`class SingleResult ():`

PLAN **`molecule`** & **`method`** & **`func`** unchanged. `return json`

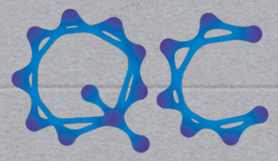
ASM Return analytic energy, gradient, or Hessian.

psi4.optimize('HF/cc-pv[d,t]z', bsse_type='cp', molecule= )

DRIV & Fractal DEMO

CP-CORR CBS OPT

```python
from qcfractal import FractalSnowflake
from qcfractal.interface import FractalClient

# Build a active server and client
snowflake = FractalSnowflake()
client = FractalClient(snowflake)
print(client)


def psi_model(coords):

    dimer = psi4.geometry("""O 0 0 0\n H 1 0 0\n H 0 1 0\n --\n O 3 3 3\n H
    dimer.update_geometry()
    dimer.set_geometry(psi4.core.Matrix.from_array(coords))

    plan = psi4.gradient("HF/cc-pV[D, T]Z", bsse_type="CP", molecule=dimer,
                         return_plan=True, return_total_data=True)

    plan.compute(client)

    snowflake.await_results()
    ret = plan.get_results(client)

    return (ret["extras"]["qcvars"]["CURRENT ENERGY"],
   np.array(ret["extras"]["qcvars"]["CURRENT GRADIENT"]).reshape(-1, 3))
```

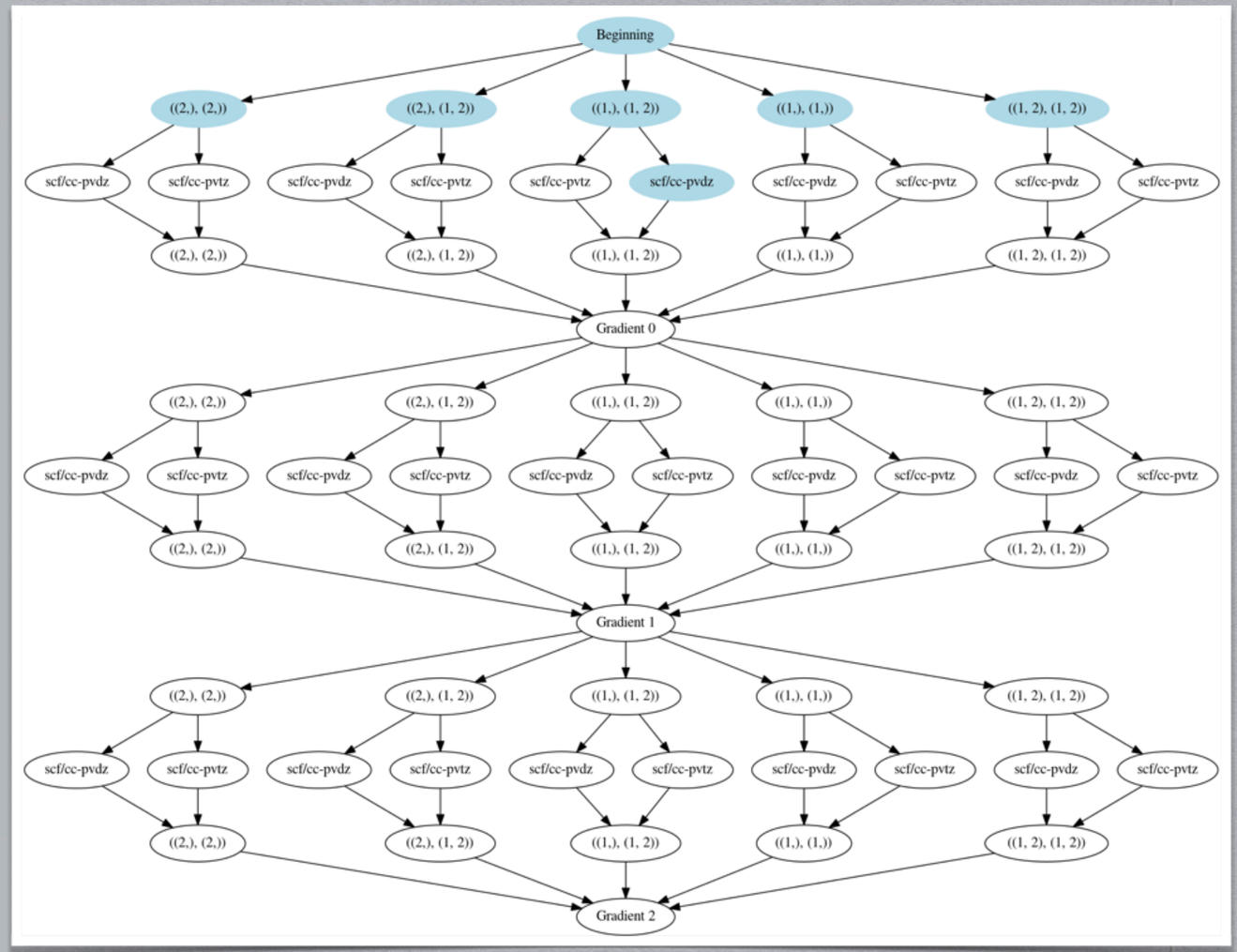


QC Archive

A MolSSI Project

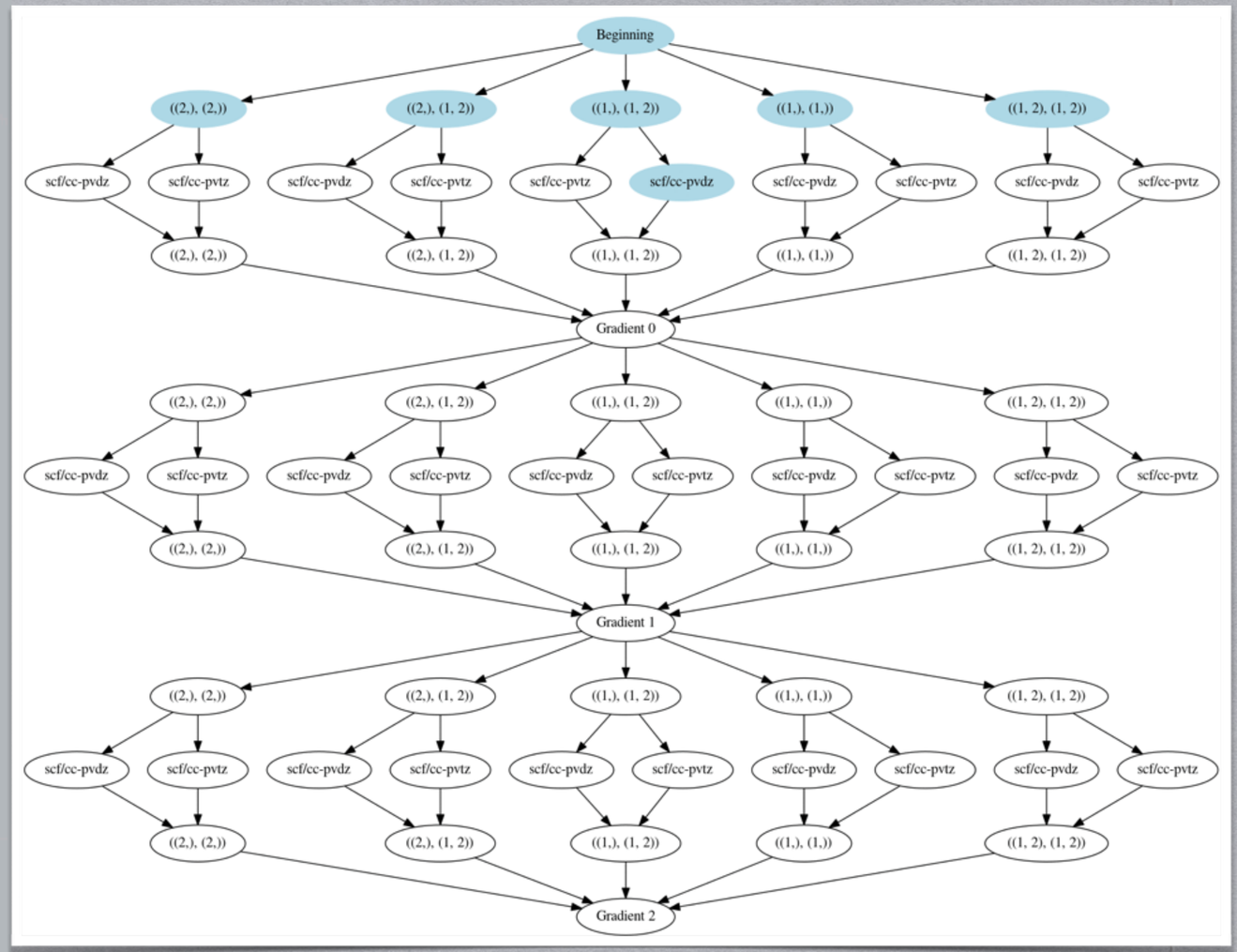
psi4.optimize('HF/cc-pv[d,t]z', bsse_type='cp', molecule=

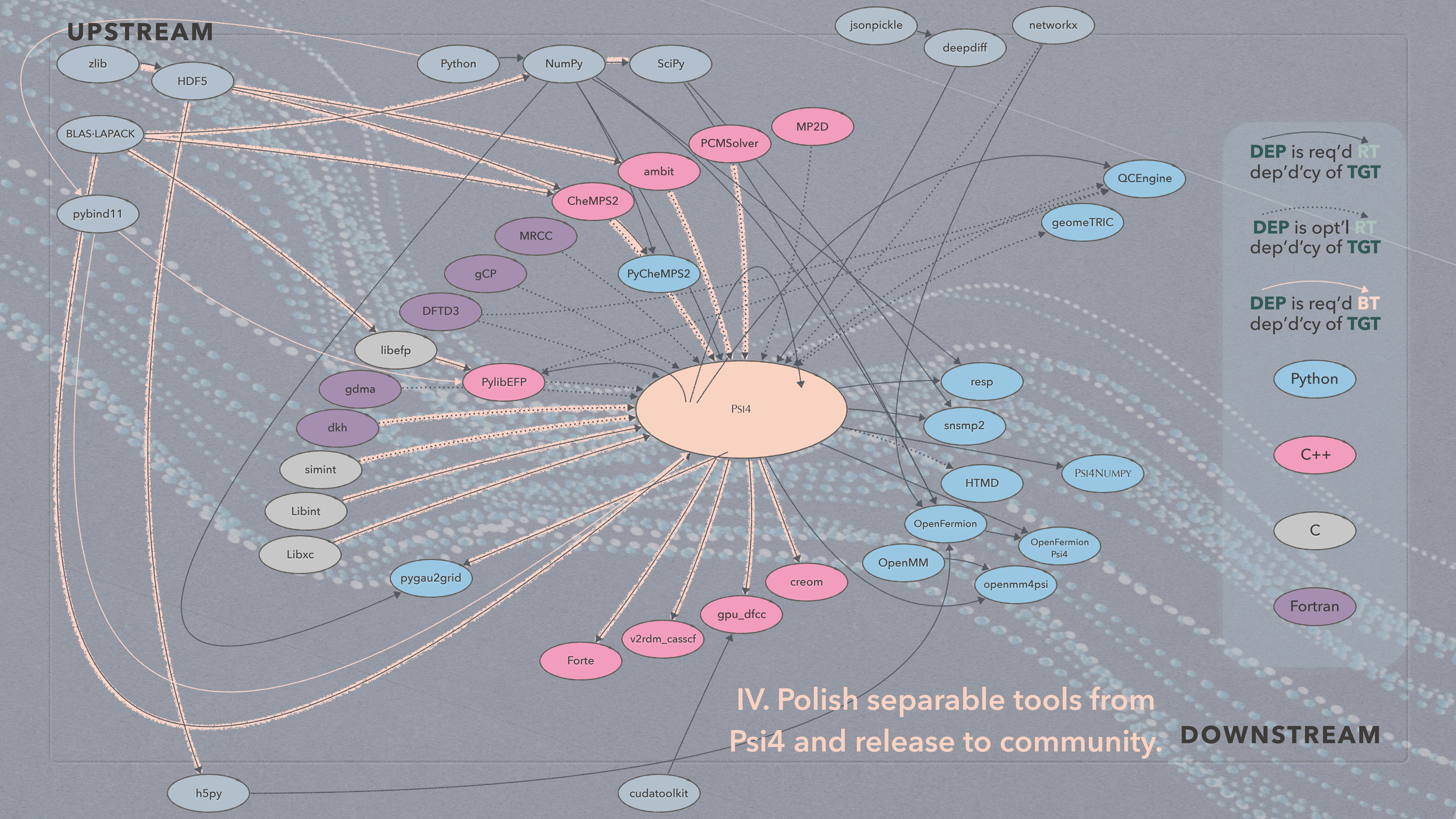
)

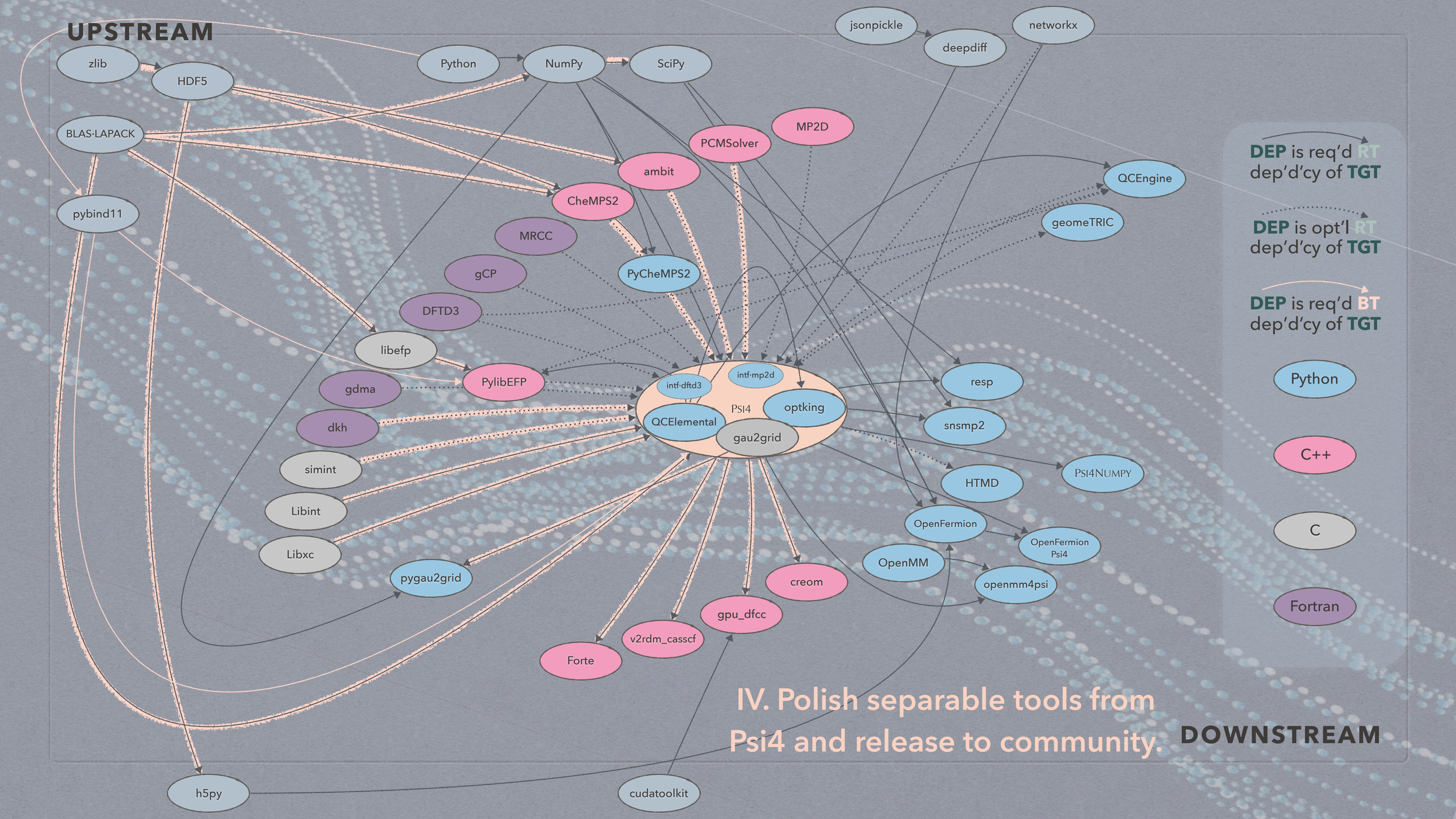


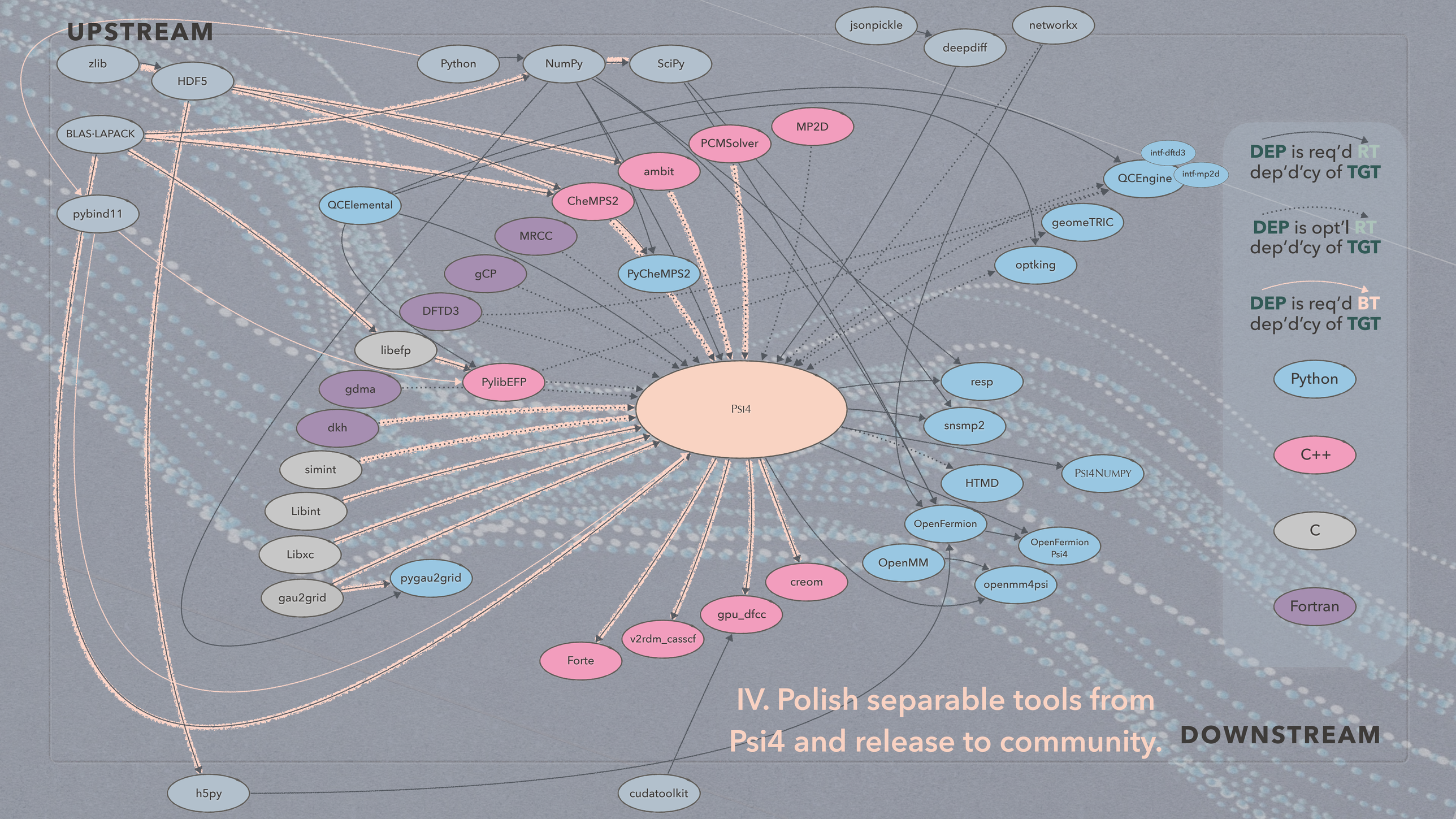
psi4.optimize('HF/cc-pv[d,t]z', bsse_type='cp', molecule=

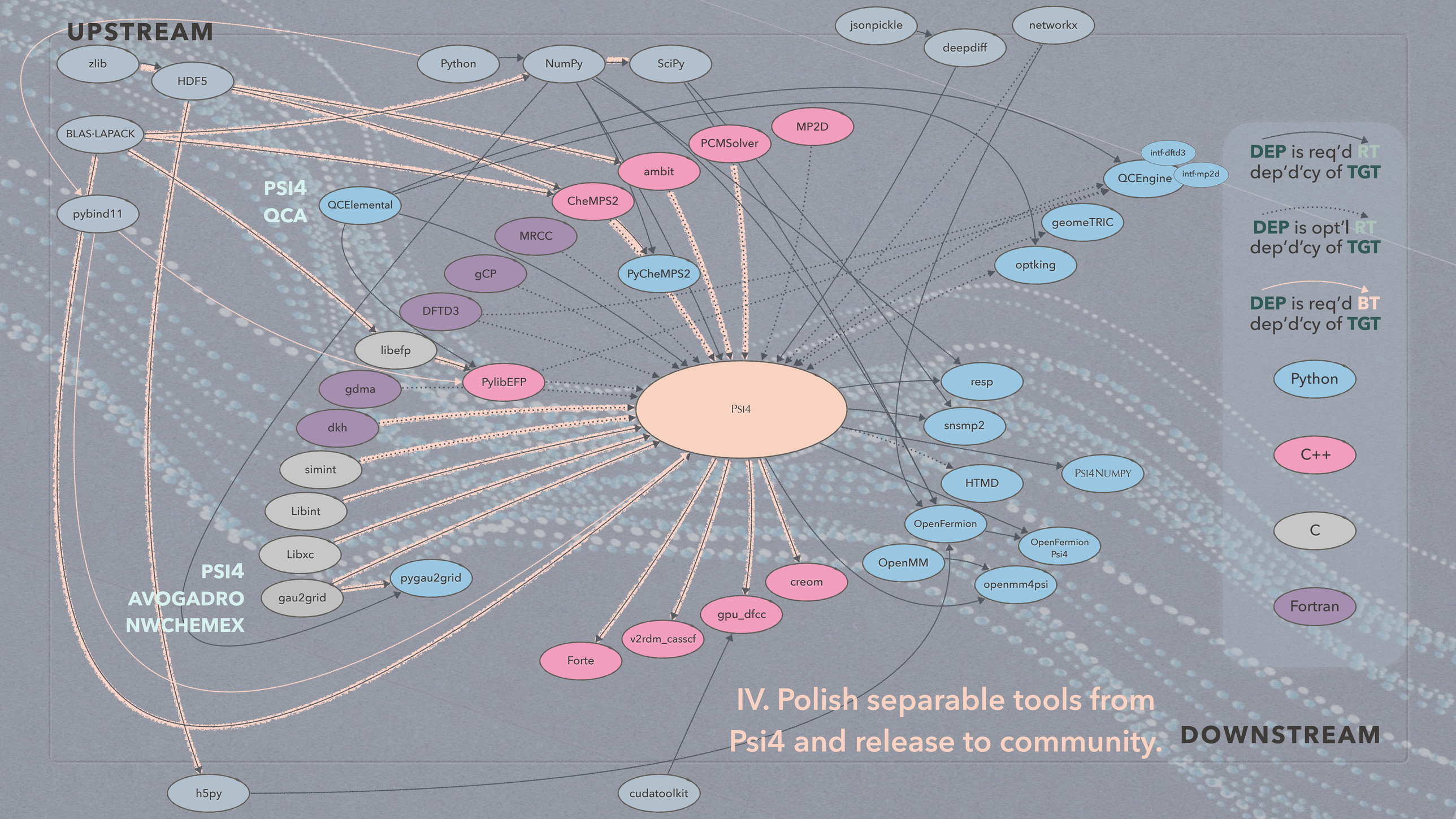
)




UPSTREAM
zlib
HDF5
Python
NumPy
SciPy
jsonpickle
deepdiff
networkx
BLAS·LAPACK
MP2D
PCMSolver
ambit
QCEngine
CheMPS2
pybind11
geomeTRIC
MRCC
gCP
PyCheMPS2
DFTD3
libefp
PylibEFP
gdma
dkh
simint
Libint
Libxc
Psi4
resp
snsmp2
Psi4NumPy
HTMD
OpenFermion
OpenFermion Psi4
OpenMM
openmm4psi
pygau2grid
creom
gpu_dfcc
v2rdm_casscf
Forte
h5py
cudatoolkit
DEP is req'd RT dep'd'cy of TGT
DEP is opt'l RT dep'd'cy of TGT
DEP is req'd BT dep'd'cy of TGT
Python
C++
C
Fortran
IV. Polish separable tools from Psi4 and release to community.
DOWNSTREAM


UPSTREAM
zlib
HDF5
Python
NumPy
SciPy
jsonpickle
deepdiff
networkx
BLAS·LAPACK
MP2D
PCMSolver
ambit
QCEngine
CheMPS2
pybind11
geomeTRIC
MRCC
gCP
PyCheMPS2
DFTD3
libefp
PylibEFP
gdma
dkh
simint
Libint
Libxc
intf·dftd3
intf·mp2d
PSI4
QCElemental
optking
gau2grid
resp
snsmp2
PSI4NUMPY
HTMD
OpenFermion
OpenFermion Psi4
OpenMM
openmm4psi
pygau2grid
creom
gpu_dfcc
v2rdm_casscf
Forte
h5py
cudatoolkit
DEP is req'd RT dep'd'cy of TGT
DEP is opt'l RT dep'd'cy of TGT
DEP is req'd BT dep'd'cy of TGT
Python
C++
C
Fortran
IV. Polish separable tools from Psi4 and release to community.
DOWNSTREAM


UPSTREAM
zlib
HDF5
Python
NumPy
SciPy
jsonpickle
deepdiff
networkx
BLAS·LAPACK
pybind11
PCMSolver
MP2D
ambit
CheMPS2
QCElemental
MRCC
gCP
PyCheMPS2
DFTD3
libefp
PylibEFP
gdma
dkh
simint
Libint
Libxc
gau2grid
pygau2grid
Psi4
creom
gpu_dfcc
v2rdm_casscf
Forte
h5py
cudatoolkit
intf·dftd3
QCEngine
intf·mp2d
geomeTRIC
optking
resp
snsmp2
Psi4NumPy
HTMD
OpenFermion
OpenFermion Psi4
OpenMM
openmm4psi
DEP is req'd RT dep'd'cy of TGT
DEP is opt'l RT dep'd'cy of TGT
DEP is req'd BT dep'd'cy of TGT
Python
C++
C
Fortran
IV. Polish separable tools from Psi4 and release to community.
DOWNSTREAM


UPSTREAM
zlib
HDF5
Python
NumPy
SciPy
jsonpickle
deepdiff
networkx
BLAS·LAPACK
pybind11
PSI4
QCA
QCElemental
CheMPS2
ambit
PCMSolver
MP2D
MRCC
gCP
DFTD3
PyCheMPS2
libefp
PylibEFP
gdma
dkh
simint
Libint
Libxc
gau2grid
pygau2grid
PSI4
AVOGADRO
NWCHEMEX
h5py
cudatoolkit
Forte
v2rdm_casscf
gpu_dfcc
creom
PSI4
intf·dftd3
intf·mp2d
QCEngine
geomeTRIC
optking
resp
snsmp2
PSI4NUMPY
HTMD
OpenFermion
OpenFermion Psi4
OpenMM
openmm4psi
DEP is req'd RT dep'd'cy of TGT
DEP is opt'l RT dep'd'cy of TGT
DEP is req'd BT dep'd'cy of TGT
Python
C++
C
Fortran
IV. Polish separable tools from Psi4 and release to community.
DOWNSTREAM

QCDB DEMO
4-PROGRAM FOCAL POINT

```
(p4dev37) psilocaluser@bash:psinet:/home/psilocaluser/gits/qcddb/demo2: (modernps12) time python zach.py
```

Name	Method	Program
E_{Base}	CCSD(T) / cc-pCV[Q5]Z	NWCHEM
ΔE_{Basis}	MP2 / [aug-cc-pCV[56]Z - cc-pCV[Q5]Z]	PSI
ΔE_{DBOC}	(at) CCSD / cc-pCVDZ	CFOUR
$\Delta E_{Relativistic}$	(at) CCSD(T) / cc-pCVTZ	PSI
ΔE_{CCSDTQ}	CCSDTQ / cc-pVTZ - CCSD(T) / cc-pVTZ	CFOUR/MRCC
ΔE_{FCI}	FCI / cc-pVDZ - CCSD(T) / cc-pVDZ	GAMESS

```
qcdb.set_options({
        'memory': '35 gb',
        'e_convergence': 1e-11,
        'scf__d_convergence': 1e-9,
        'nwchem_ccsd__maxiter': 100,
        'psi4_mp2_type': 'conv',
        'psi4_scf_type': 'direct',
        'psi4_df_scf_guess': 'false',
                    })
```

```
# ccsdtq correction: (CCSDTQ - CCSD(T)) / cc-pVDZ
qcdb.set_options({'cfour_dropmo': [1],})
_, jrec = qcdb.energy('c4-ccsd(t)/cc-pVTZ', return_wfn=True)
E_ccsdpt = float(jrec['qcvars']['CCSD(T) TOTAL ENERGY'].data)
_, jrec = qcdb.energy('c4-ccsdtq/cc-pVTZ', return_wfn=True)
E_ccsdtq = float(jrec['qcvars']['CCSDTQ TOTAL ENERGY'].data)
qcdb.set_options({'cfour_dropmo': None})
dE_ccsdtq[i] = E_ccsdtq - E_ccsdpt
print(f'~~~ CCSDTQ Correction={dE_ccsdtq[-1]} Har. ({i+1}/{npoints}) ~~~')

# base calculation: CCSD(T) / cc-pCV[Q5]Z
qcdb.set_options({'memory': '10 gb'})
#E, jrec = qcdb.energy('nwc-ccsd(t)/cc-pCV[T,Q]Z', return_wfn=True)
E, jrec = qcdb.energy('nwc-ccsd(t)/cc-pCVTZ', return_wfn=True)
qcdb.set_options({'memory': '55 gb'})
E_base[i] = E
print(f'~~~ Base Energy={E} Har. ({i+1}/{npoints}) ~~~')

# basis set correction: MP2 / (aug-cc-pCV[56]Z) - cc-pCV[Q5]Z)
E_small, _ = qcdb.energy('p4-mp2/cc-pCV[T,Q]Z', return_wfn=True)
E_large, _ = qcdb.energy('p4-mp2/aug-cc-pCV[T,Q]Z', return_wfn=True)
dE_basis[i] = E_large - E_small
print(f'~~~ Basis Correction={dE_basis[-1]} Har. ({i+1}/{npoints}) ~~~')

# relativistic correction: (X2C-CCSD(T) - CCSD(T)) / cc-pCVTZ-DK
qcdb.set_options({'psi4_relativistic': 'x2c'})
E_x2c_on, jrec = qcdb.energy('p4-ccsd(t)/aug-cc-pCVTZ-DK', return_wfn=True)
qcdb.set_options({'psi4_relativistic': 'no'})
E_x2c_off, jrec = qcdb.energy('p4-ccsd(t)/aug-cc-pCVTZ-DK', return_wfn=True)
dE_x2c[i] = E_x2c_on-E_x2c_off
print(f'~~~ Relativistic Correction={dE_x2c[-1]} Har. ({i+1}/{npoints}) ~~~')

# fci correction: (FCI - CCSD(T)) / cc-pVDZ
E_cc, _ = qcdb.energy('gms-ccsd(t)/cc-pVDZ', return_wfn=True)
E_fci, _ = qcdb.energy('gms-fci/cc-pVDZ', return_wfn=True)
dE_fci[i] = E_fci - E_cc
print(f'~~~ FCI Correction={dE_fci[-1]} Har. ({i+1}/{npoints}) ~~~')
```

```
phys_consts_tot_fci = psi4.diatomic.anharmonicity(R_arr, E_tot_fci)
```

QCDB DEMO

4-PROGRAM FOCAL POINT

```
(p4dev37) psilocaluser@bash:psinet:/home/psilocaluser/gits/qccddb/demo2: (modernpsi2) time python zach.py
```

Psi4

Jerome Gonthier
Berkeley

Rob Parrish
Stanford

Ryan Richard
Iowa State

Rollin King
Bethel

Alexander Sokolov
Ohio State

David Sherrill
GaTech

Lori Burns
GaTech

Francesco Evangelista
Emory

Eugene DePrince
FSU

Fritz Schaefer
UGA

Justin Turney
UGA

Daniel Crawford
VaTech

Daniel Smith
MolSSI

Ben Pritchard
MolSSI

Andrew James
VaTech

Ed Valeev
VaTech

Andy Simmonett
NIH

Ed Hohenstein
CCNY

Roberto Di Remigio
Tromso

Ugur Bozkaya
Hacettepe

QCA & QCDB ACKNOWLEDGEMENTS

QCArchive

Daniel Smith
MolSSI

Levi Naden
MolSSI

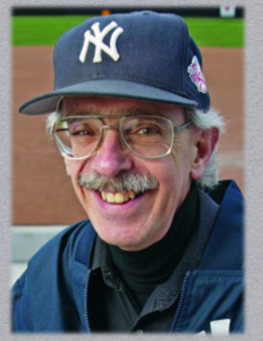
NWChem

Theresa Windus
Iowa State

Jiyoung Lee
Iowa State

Annabelle Lolinco
Iowa State

GAMESS



Mark Gordon
Iowa State

Nuwan de Silva

CFOUR

John Stanton
UFL

Devin Matthews
UTexas

Psi4

David Sherrill
GaTech

Daniel Smith
MolSSI

DEMOs

Zach Glick
GaTech

Asim Alenaizan
GaTech

Zach Glick
GaTech

QCA & QCDB ACKNOWLEDGEMENTS

QCArchive

Daniel Smith
MolSSI

Levi Naden
MolSSI

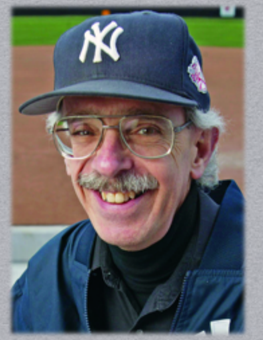
NWChem

Theresa Windus
Iowa State

Jiyoung Lee
Iowa State

Annabelle Lolinco
Iowa State

GAMESS



Mark Gordon
Iowa State

Nuwan de Silva

CFOUR

John Stanton
UFL

Devin Matthews
UTexas

Psi4

David Sherrill
GaTech

Daniel Smith
MolSSI

SHERRILL GROUP

QCA & QCDB ACKNOWLEDGEMENTS

Daniel Smith
MolSSI

Levi Naden
MolSSI

Theresa Windus
Iowa State

Ben Pritchard
MolSSI

QCA & QCDB ACKNOWLEDGEMENTS

Daniel Smith
MolSSI

Levi Naden
MolSSI

Theresa Windus
Iowa State

Ben Pritchard
MolSSI

MolSSI