

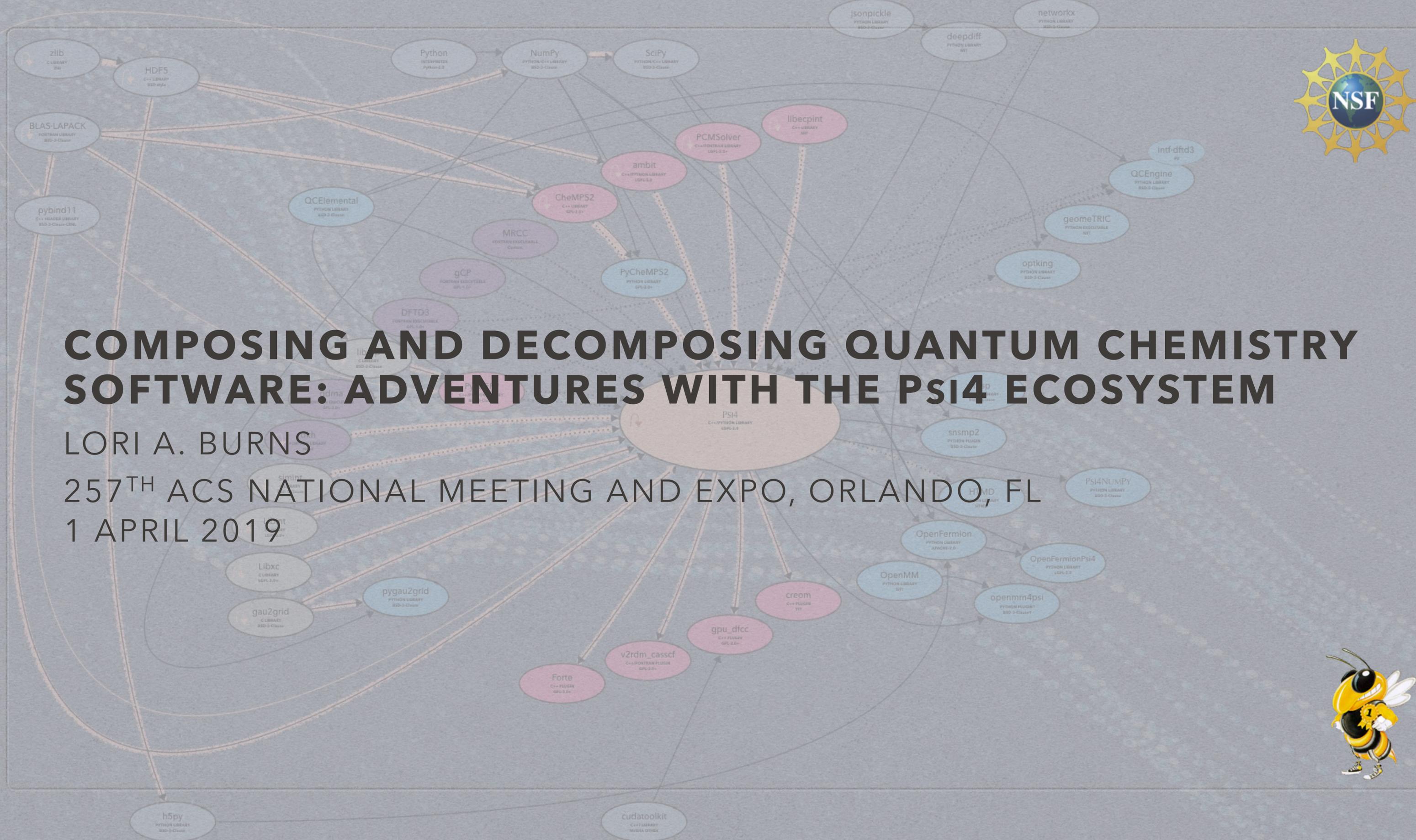


COMPOSING AND DECOMPOSING QUANTUM CHEMISTRY SOFTWARE: ADVENTURES WITH THE Psi4 ECOSYSTEM

LORI A. BURNS

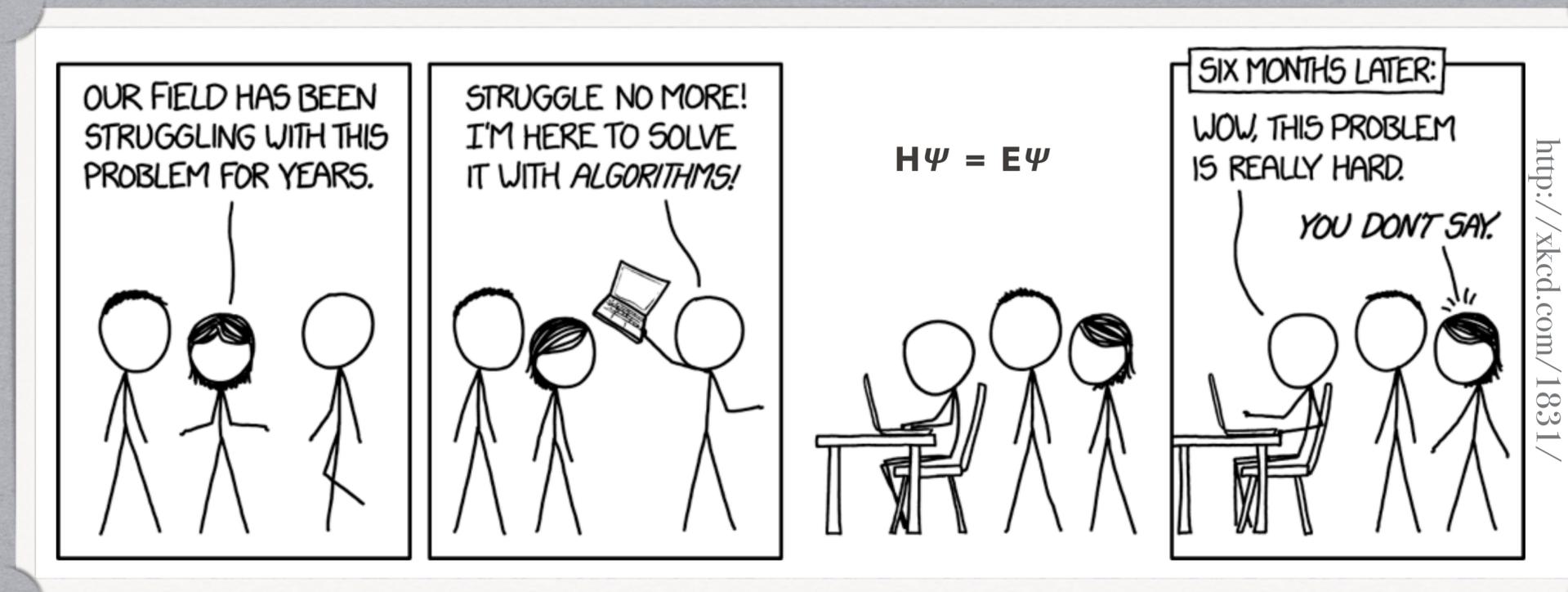
257TH ACS NATIONAL MEETING AND EXPO, ORLANDO, FL

1 APRIL 2019



QUANTUM CHEMISTRY HAS PROBLEMS ENOUGH

LET OTHERS SOLVE THE REST



QUANTUM CHEMISTRY HAS PROBLEMS ENOUGH LET OTHERS SOLVE THE REST

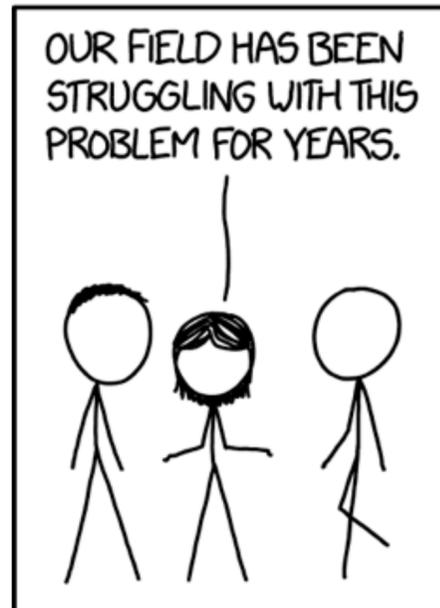
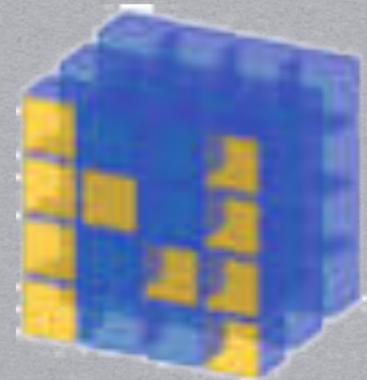


python



CMake

pybind11



<http://xkcd.com/1831/>



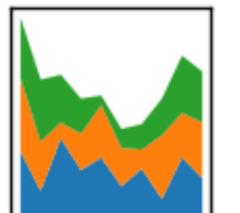
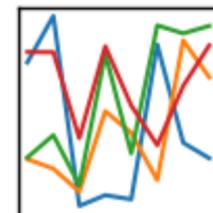
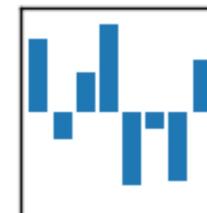
L	A	P	A	C	K
L	-A	P	-A	C	-K
L	A	P	A	-C	-K
L	-A	P	-A	-C	K
L	A	-P	-A	C	K
L	-A	-P	A	C	-K



git

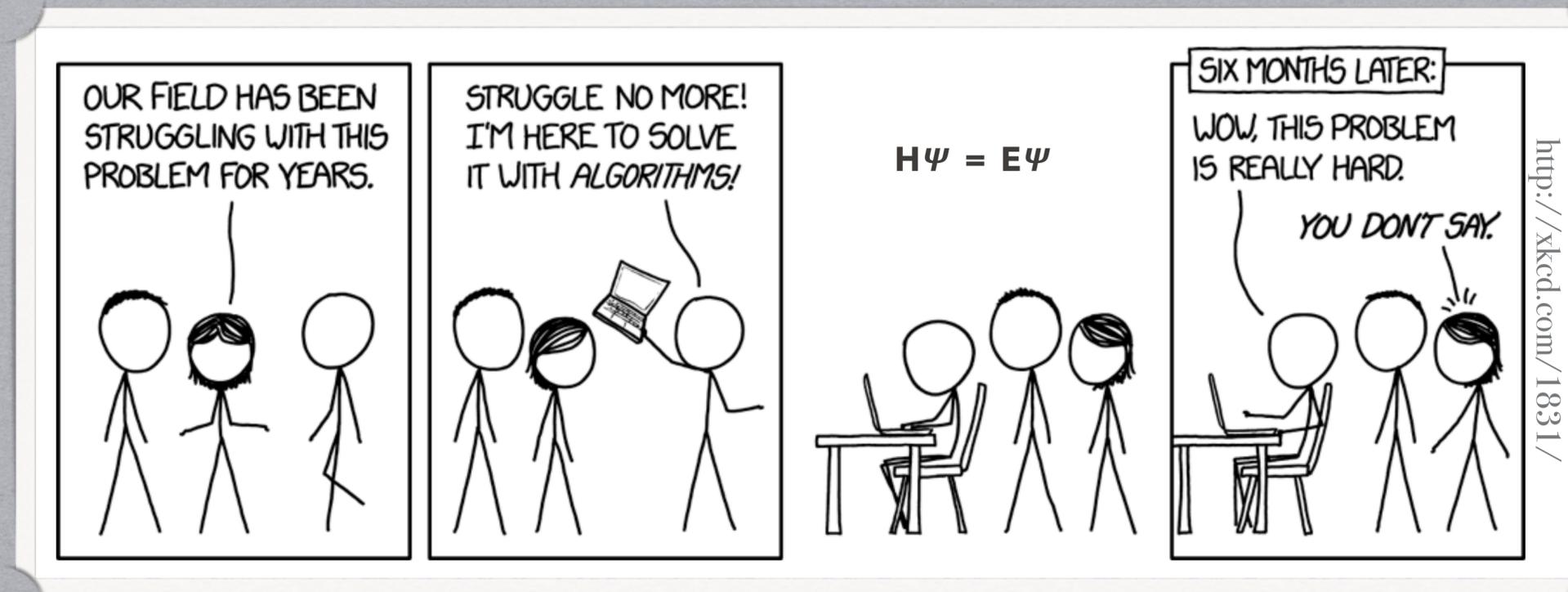
pandas

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



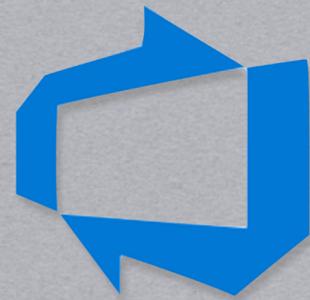
QUANTUM CHEMISTRY HAS PROBLEMS ENOUGH

LET OTHERS SOLVE THE WORKFLOW



QUANTUM CHEMISTRY HAS PROBLEMS ENOUGH

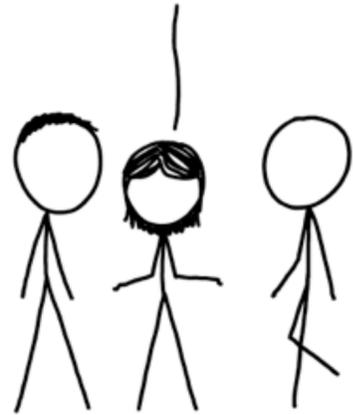
LET OTHERS SOLVE THE WORKFLOW



ANACONDA
CLOUD



OUR FIELD HAS BEEN STRUGGLING WITH THIS PROBLEM FOR YEARS.



STRUGGLE NO MORE!
I'M HERE TO SOLVE
IT WITH *ALGORITHMS!*



$$H\psi = E\psi$$



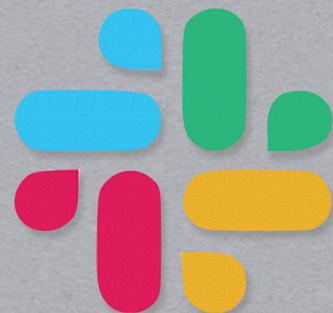
SIX MONTHS LATER:

WOW, THIS PROBLEM
IS REALLY HARD.

YOU DON'T SAY.



<http://xkcd.com/1831/>



CONDA-FORGE

GIT WORKFLOW

UNGUIDED TO GUIDED

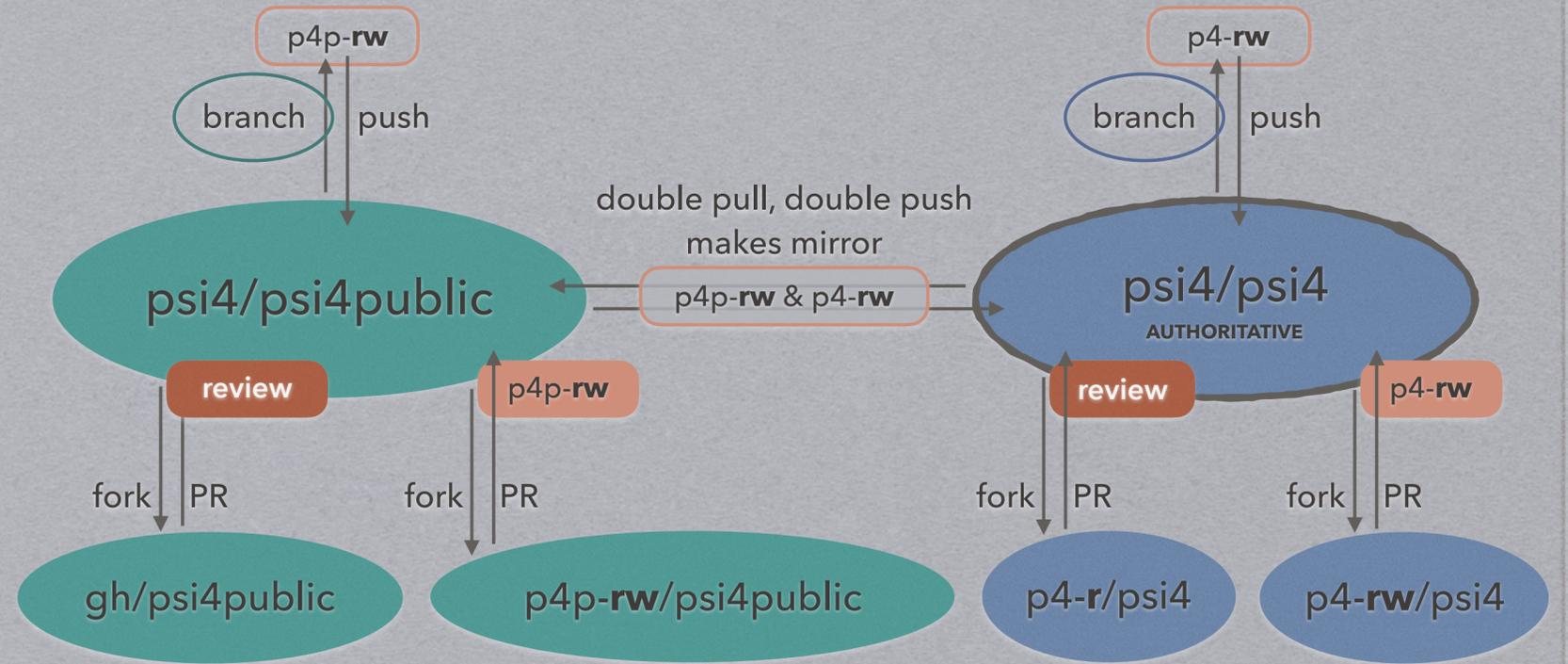
public

private

GIT WORKFLOW

UNGUIDED TO GUIDED

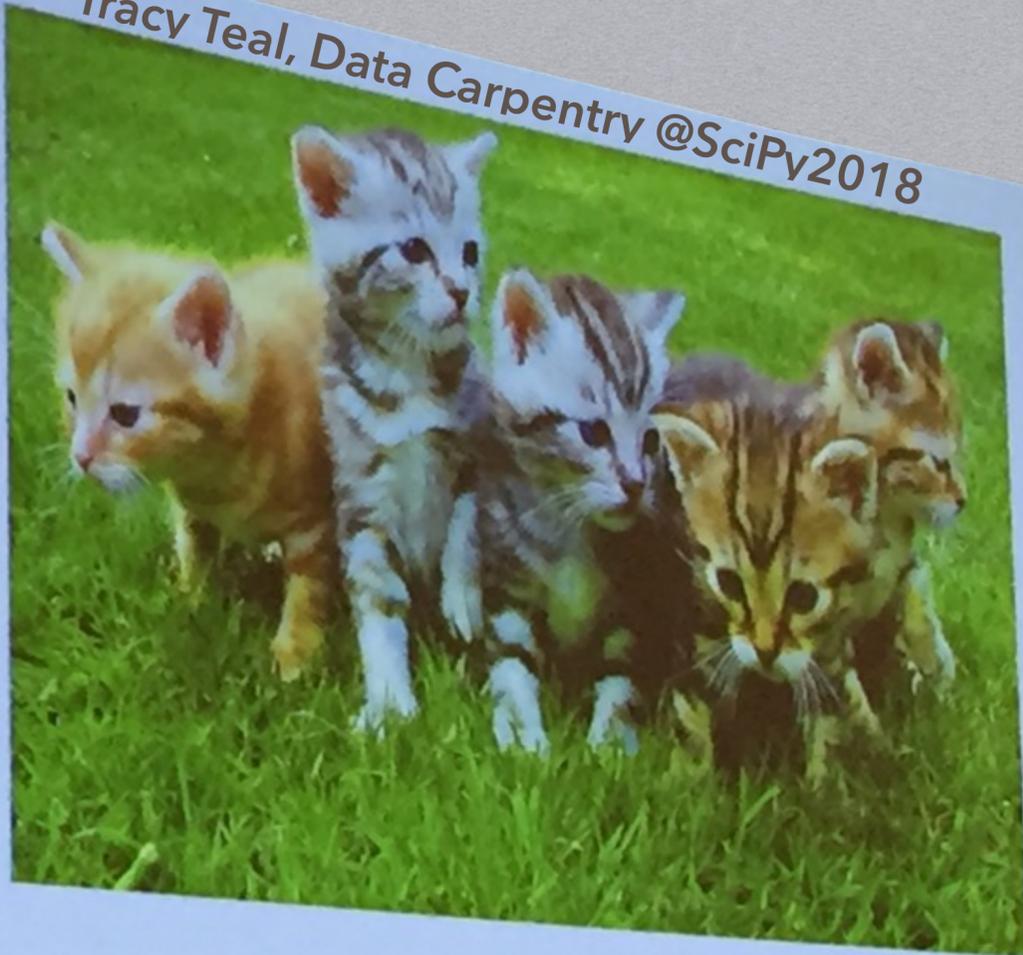
2015



GIT WORKFLOW

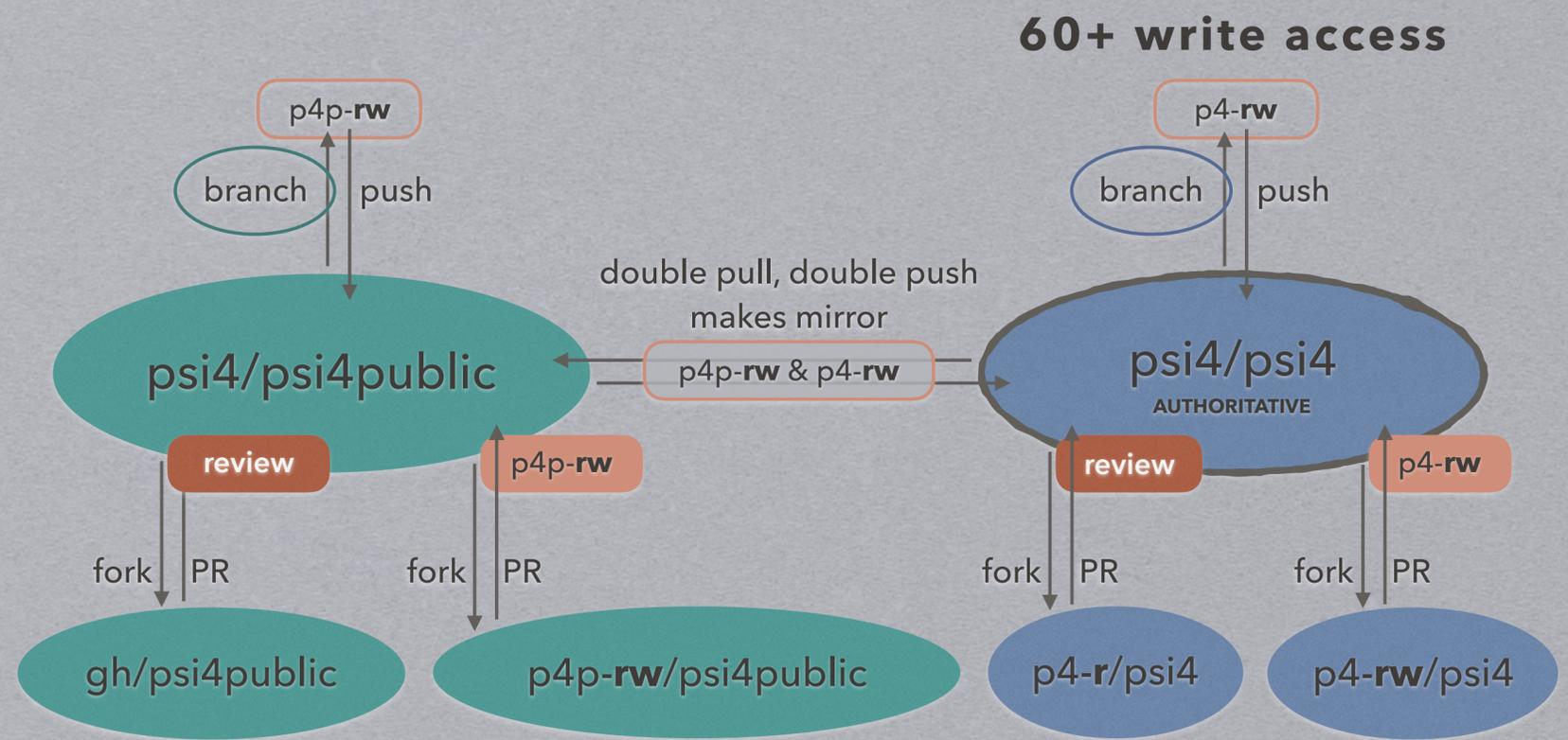
UNGUIDED TO GUIDED

Tracy Teal, Data Carpentry @SciPy2018



If you want to go fast, go alone.
If you want to go in a lot of different
uncoordinated directions at once, go together.

2015



public

private

GIT WORKFLOW

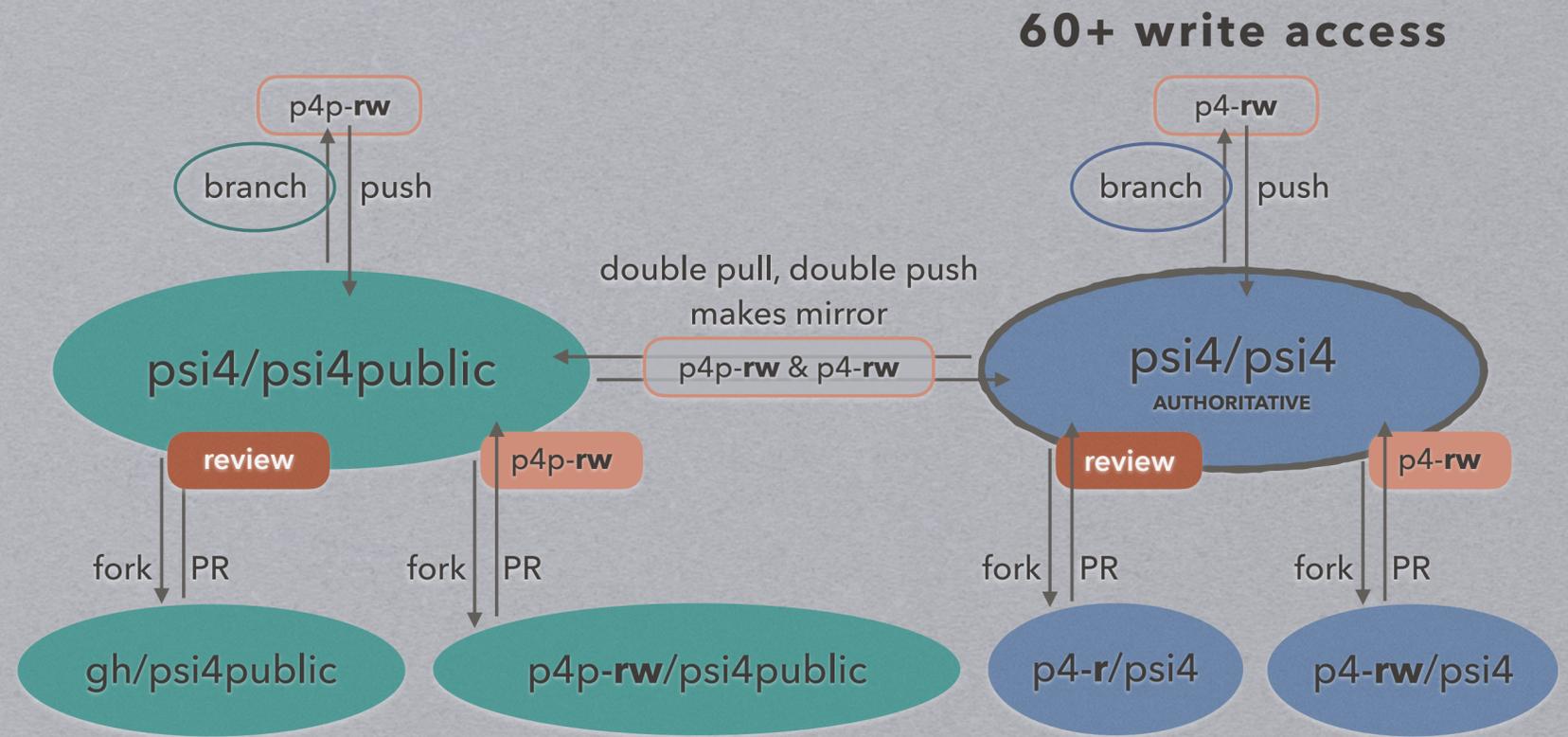
UNGUIDED TO GUIDED

Tracy Teal, Data Carpentry @SciPy2018

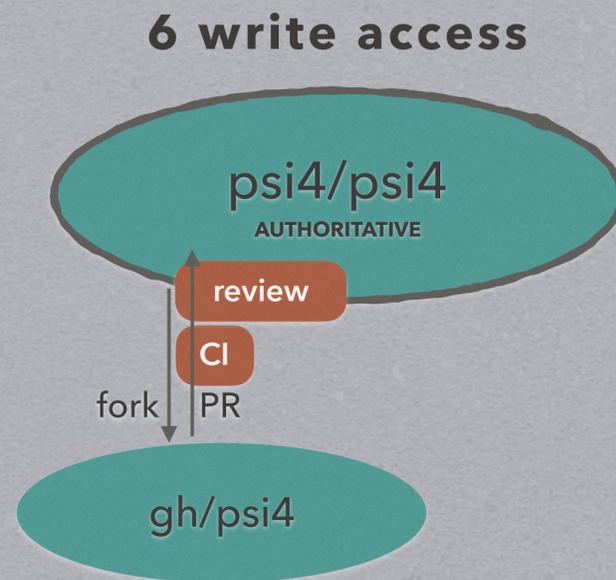


If you want to go fast, go alone.
If you want to go in a lot of different uncoordinated directions at once, go together.

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"

BUS FACTOR

RESILIANCY OF PROJECT



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

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

BUS FACTOR

RESILIANCY OF PROJECT



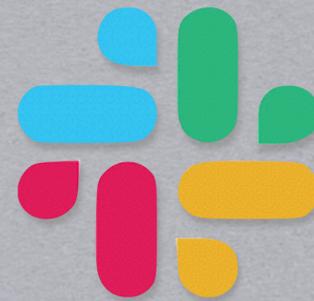
**NUMBER OF DEVELOPERS WHO,
IF SUDDENLY ABSENT, STALL A PROJECT**

- **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

COMMUNICATION

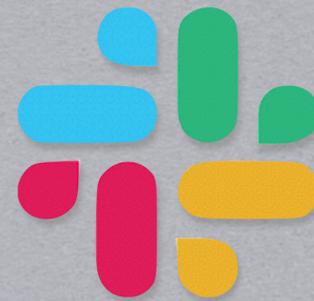
GITHUB AND SLACK



- **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



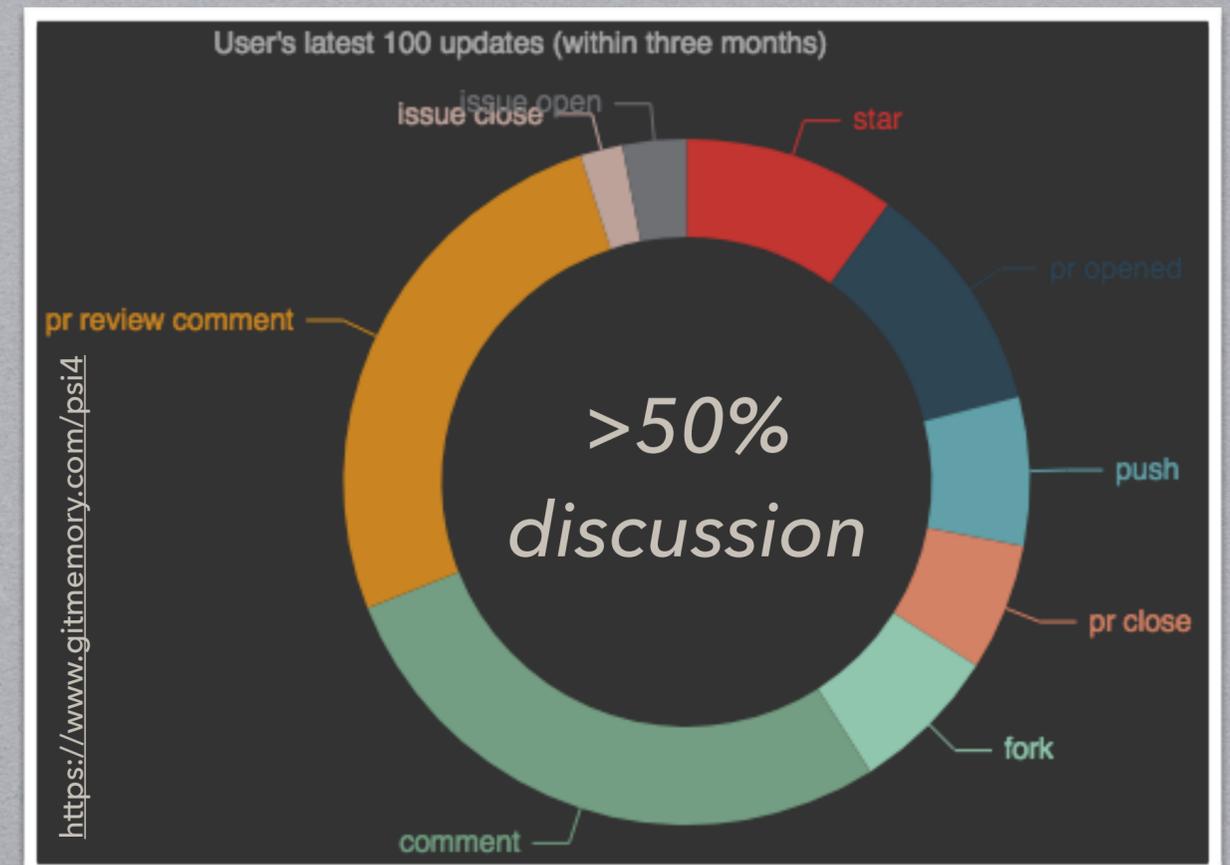
Psi4

psi4.slack.com

Your workspace is currently on Slack's free plan. [See upgrade options](#)

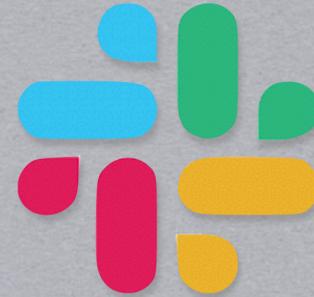
Total Messages

Upgrade to access your first 44.6k messages.



COMMUNICATION

GITHUB AND SLACK



 dev chat on 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.



Psi4

psi4.slack.com



Jonathon Misiewicz Mar 2nd at 12:43 PM

Neither - trying to get IWL out of `dcft`. `libtrans` insists on the TPDM being in an IWL buffer. So every non-SCF gradient code needs to dump their `libdpd` densities into `libiwl` buffers... so that `libtrans` can turn them into `libdpd` buffers. This seems overcomplicated. I'm currently trying to figure out what the filler functors are doing. It was a major epiphany that it is not creating the TPDM in the new buffers, but adding TPDM elements together for ease of contraction with the derivative integrals.

1 reply



andysim 15 hours ago

`libtrans` was written to glue together the `psi3` modules, particularly the CI and CC codes, as we transitioned to a single-executable model, which became `psi4`, they both provided IWL output for the TPMD, so that's what I went with as input. if you have the DPD form already in DCFT, you can add some logic to DCFT to override the IWL step



hokru 10:41 AM

Where did the `dft-bench-interaction` test go?



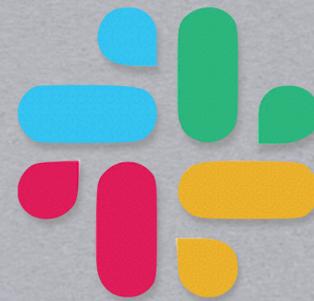
loriab 10:42 AM

https://github.com/psi4/psi4/blob/master/tests/pytest/test_dft_benchmarks.py

<https://www.gitmemory.com/psi4>

COMMUNICATION

GITHUB AND SLACK



dev chat on 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.

 **Psi4**
psi4.slack.com

 **Jonathon Misiewicz** Mar 2nd at 12:43 PM
Neither - trying to get IWL out of `dcft`.
`libtrans` insists on the TPDM being in an IWL buffer. So every non-SCF gradient code needs to dump their `libdpd` densities into `libiwl` buffers... so that `libtrans` can turn them into `libdpd` buffers. This seems overcomplicated. I'm currently trying to figure out what the filler functors are doing.

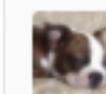
loriab 11:56 AM
I'm pleased to report that our new run-time deprecation warnings are going to be annoying enough to get everyone to update their syntax really quick.



but
base of
ls.

pr review comm

<https://www.gitmemory.com/psi4>

 **andysim** 15 hours ago
libtrans was written to glue together the psi3 modules, particularly the CI and CC codes, as we transitioned to a single-executable model, which became psi4, they both provided IWL output for the TPMD, so that's what I went with as input. if you have the DPD form already in DCFT, you can add some logic to DCFT to override the IWL step

 **hokru** 10:41 AM
Where did the `dft-bench-interaction` test go?

 **loriab** 10:42 AM
https://github.com/psi4/psi4/blob/master/tests/pytest/test_dft_benchmarks.py

CONTINUOUS INTEGRATION

TRAVIS-CI & AZURE



CONTINUOUS INTEGRATION

TRAVIS-CI & AZURE



MAC: 11, WINDOWS: 1, LINUX: 1



CONTINUOUS INTEGRATION

TRAVIS-CI & AZURE

MAC: 11, WINDOWS: 1, LINUX: 1



TRAVIS-CI: LINUX[, MAC]



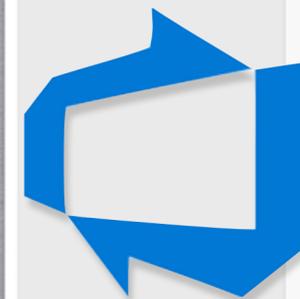
AZURE: WIN[, LINUX, MAC]

Azure DevOps interface showing a successful build job for 'psi4'.

Build Job Summary:

- Build Job: #20190322.2: Merge pull request #1578 fro...
- Agent: Hosted Agent
- Started: 3/22/2019, 5:06:06 PM
- Duration: 1h 7m 36s

Step	Status	Duration
Prepare job	succeeded	<1s
Initialize job	succeeded	2s
Checkout	succeeded	20s
Check Python version: 3.6	succeeded	2s
Check cmake configuration	succeeded	1s
Check ctest configuration	succeeded	1s
Check pytest configuration	succeeded	1s
Check conda configuration	succeeded	1s
Install Chocolatey	succeeded	13s
Install Miniconda	succeeded	1m 2s
Configure Miniconda	succeeded	3s
Install conda packages	succeeded	2m 29s
Install LLVM	succeeded	50s
Install Intel OpenMP import library	succeeded	1s
Configure Psi4	succeeded	38s
Build Psi4	succeeded	26m 13s
Install Psi4	succeeded	20s
Test Psi4 (OpenMP)	succeeded	5s
Test Psi4 (ctest quick)	succeeded	22m 19s
Test Psi4 (pytest quick)	succeeded	12m 46s
Build Psi4 package	skipped	
Publish Psi4 package	skipped	
Post-job: Checkout	succeeded	<1s
Finalize Job	succeeded	<1s



Travis CI build logs showing successful builds for clang C++ and gcc C++ compilers.

CONTINUOUS INTEGRATION

TRAVIS-CI & AZURE

loriab added some commits 6 hours ago

- ci: exonerating efp Verified ✗ 601d990
- ci: gfortran? Verified ✗ c6b011a
- ci: gfortran, take 2 Verified ✗ 8e71b37
- ci: kill fortran Verified ✗ 8b6204c
- ci: kill fortran dead Verified ✗ 7407ed3
- Update .travis.yml Verified ✓ 2e3918b

MolSSI / [cookiecutter-cms](#)  **COOKIECUTTER** Unwatch 14 Unstar 73 Fork 17

[Code](#) [Issues 7](#) [Pull requests 0](#) [Projects 0](#) [Wiki](#) [Insights](#)

Python-centric Cookiecutter for Molecular Computational Chemistry Packages

[130 commits](#) [2 branches](#) [0 releases](#) [7 contributors](#) [MIT](#)

TRAVIS-CI: LINUX[, MAC]

✓ # 3892.1	🔗	🔗 Compiler: clang C++	🔗 CXX_COMPILER='clang++-3.6' PYTHON_VER='3.6' C_COMPILER='clang-3.6' Fortran_COM	🕒 43 m
✓ # 3892.2	🔗	🔗 Compiler: gcc C++	🔗 CXX_COMPILER='g++' PYTHON_VER='3.7' C_COMPILER='gcc' Fortran_COMPILER='gfortr	🕒 48 m



 build passing  build passing

AZURE: WIN[, LINUX, MAC]

Azure DevOps psi4 / psi4 / Pipelines / Builds / psi4.psi4 / #20190322.2 Search

psi4

Overview

Pipelines

Builds

Releases

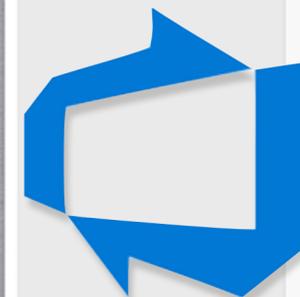
#20190322.2: Merge pull request #1578 fro... [All logs](#)

Triggered fri at 5:04 pm for dgasmith psi4/psi4 maste

Logs Summary Tests

Build Job Started: 3/22/2019, 5:06:06 PM Agent: Hosted Agent ... 1h 7m 36s

- Prepare job · succeeded <1s
- Initialize job · succeeded 2s
- Checkout · succeeded 20s
- Check Python version: 3.6 · succeeded 2s
- Check cmake configuration · succeeded 1s
- Check ctest configuration · succeeded 1s
- Check pytest configuration · succeeded 1s
- Check conda configuration · succeeded 1s
- Install Chocolatey · succeeded 13s
- Install Miniconda · succeeded 1m 2s
- Configure Miniconda · succeeded 3s
- Install conda packages · succeeded 2m 29s
- Install LLVM · succeeded 50s
- Install Intel OpenMP import library · succeeded 1s
- Configure Psi4 · succeeded 38s
- Build Psi4 · succeeded 26m 13s
- Install Psi4 · succeeded 20s
- Test Psi4 (OpenMP) · succeeded 5s
- Test Psi4 (ctest quick) · succeeded 22m 19s
- Test Psi4 (pytest quick) · succeeded 12m 46s
- Build Psi4 package · skipped
- Publish Psi4 package · skipped
- Post-job: Checkout · succeeded <1s
- Finalize Job · succeeded <1s



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



```
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      }
```

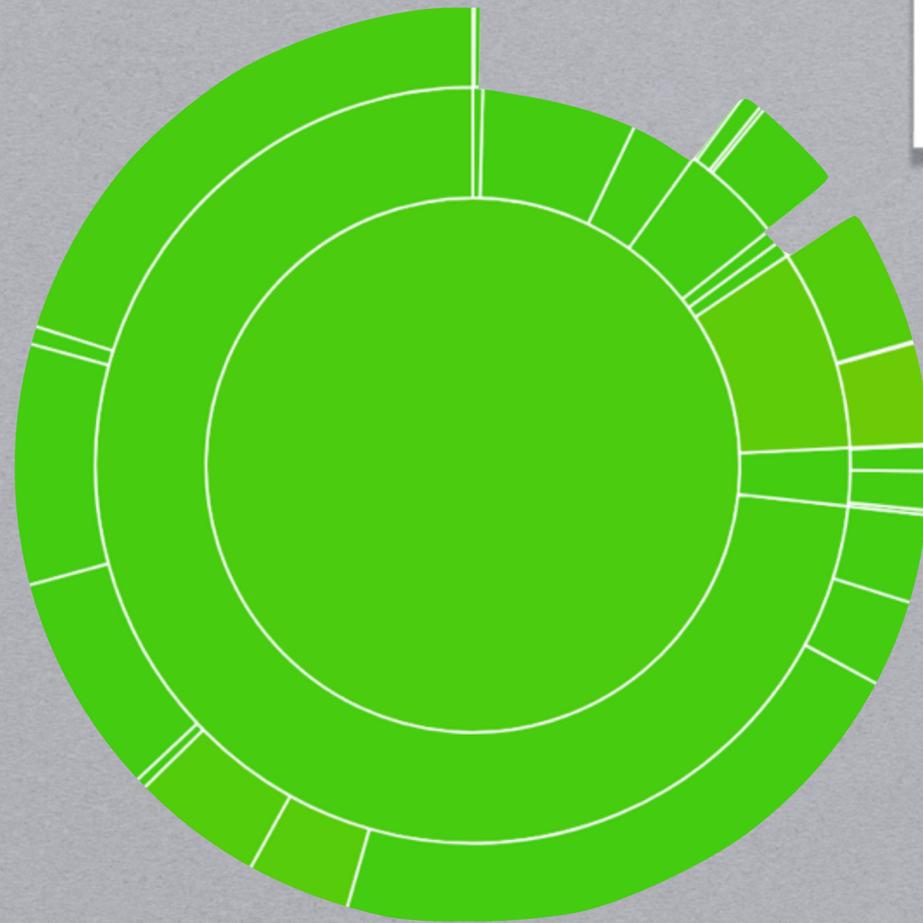
- **BUILDS** and runs test suite with different tools for different languages. Upload line tabulations to codecov site for analysis.
- **ALERTS** on lines never hit by test suite.
- **ERROR TESTING** is common area to fall short.
- **GITHUB** integration as PR check available.
- **LOCATED** bugs by mending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

DYNAMIC ANALYSIS

CODECOV



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      }
```

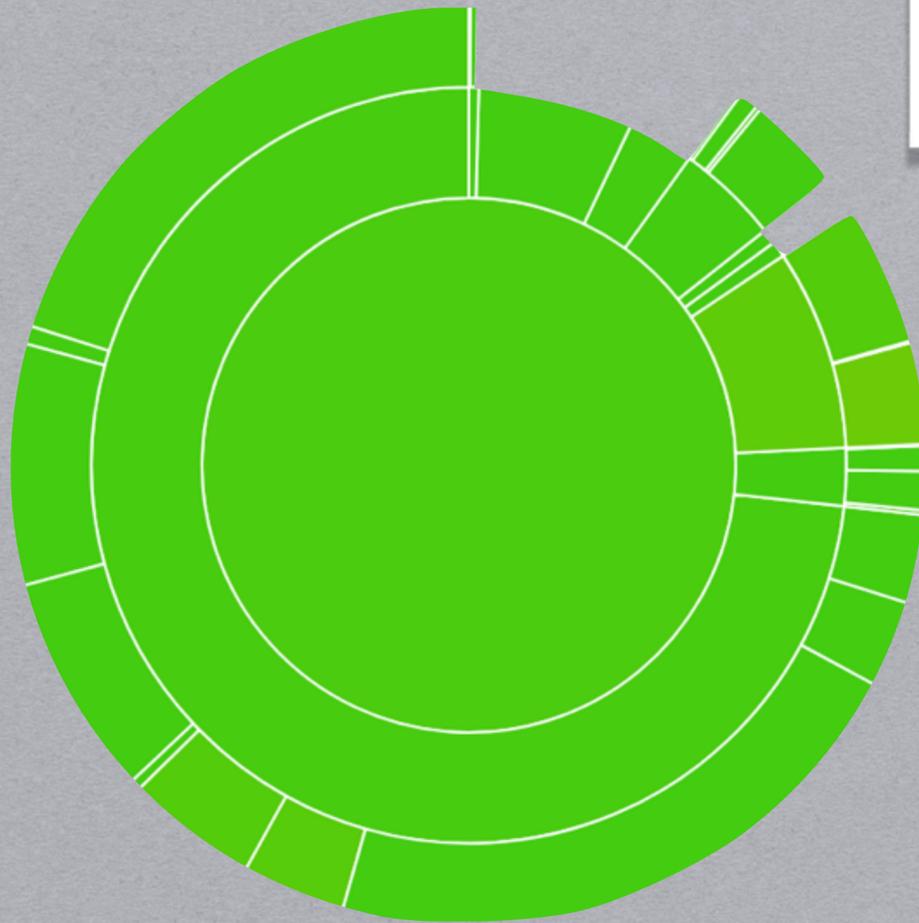
- **BUILDS** and runs test suite with different tools for different languages. Upload line tabulations to codecov site for analysis.
- **ALERTS** on lines never hit by test suite.
- **ERROR TESTING** is common area to fall short.
- **GITHUB** integration as PR check available.
- **LOCATED** bugs by mending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

DYNAMIC ANALYSIS

CODECOV

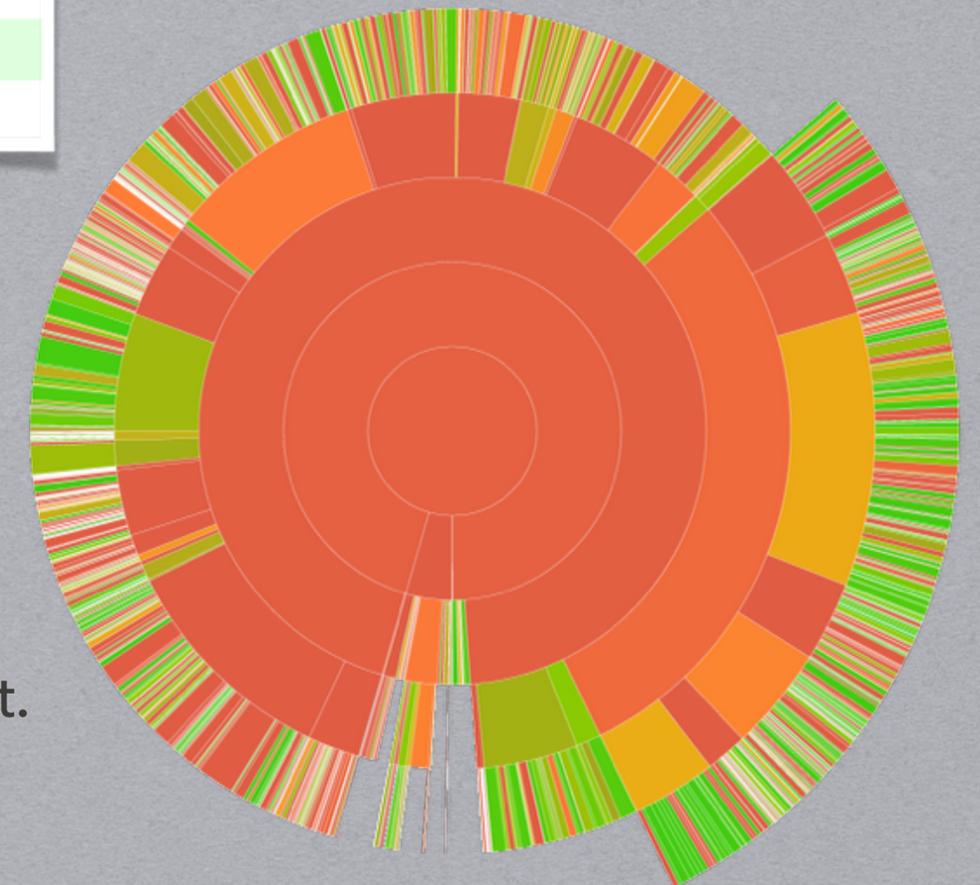


QCElemental, 99.6%



```
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 }
```

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.

 coverage 71%

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** committing wrong code twice.

STATIC ANALYSIS

LOOKS GOOD TO ME



 code quality: python **A**

```
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/*
```

- **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** committing 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** committing 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)
426
427             norm_vi = np.sqrt(engine.vector_dot(vi, vi))
428             if norm_vi >= thresh:
429                 U.append(engine.vector_scale(1.0 / norm_vi, vi))
430
431     return U
```

The value assigned to local variable 'Vi' is never used.

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);
67
68     auto tmp = block_matrix(nvir, nso);
69     auto tot1 = 0;
↓ 69-153
```

Multiplication result may overflow 'int' before it is converted to 'size_t'.

Multiplication result may overflow 'int' before it is converted to 'size_t'.

Displaying 1937 alerts, ordered by significance. 

211 Errors

1322 Warnings

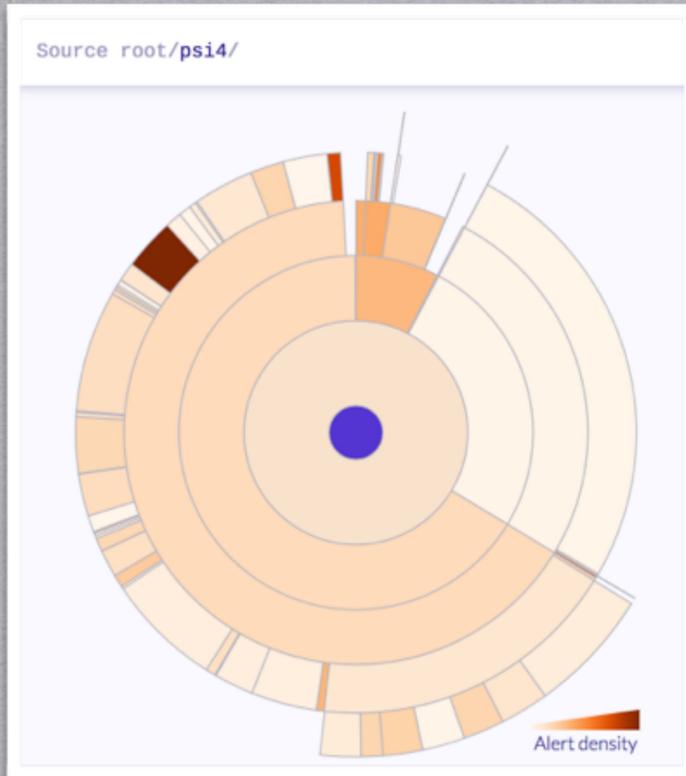
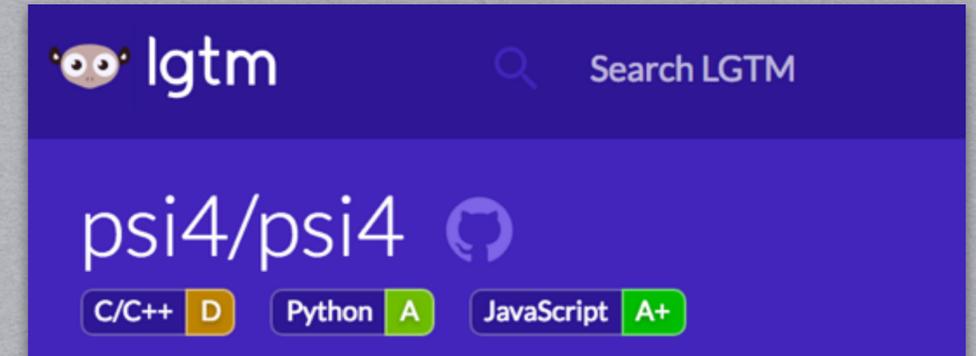
404 Recommendations

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** committing wrong code twice.

core_dev
amjames and you

loriab 21 hours ago
@amjames take a look at
<https://lgtm.com/projects/g/psi4/psi4/snapshot/879dbbf8215b932e98217a28e0f0df29274a07b0/files/psi4/driver/p4util/solvers.py?sort=name&dir=ASC&mode=heatmap#xc3e86542e4e0292:1>

amjames 4 minutes ago
Sorry for the delay, I had to duck out early yesterday. That line is an oversight/wrong, it causes no bugs because the only implementations of `engine` so far do an update to the 3rd arg and return a reference to it `b += a*x; return b`. If someone were to write an engine that returns a new object holding `b + a *x` that `_gs_orth` function wouldn't work.

loriab < 1 minute ago
good catch for lgtm, then. when you get a chance, please patch it up.

👍 1

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/>

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/>

COMPROMISE

- **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.

- **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**

- **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.



This is not a apology,
this is a warning:

If it's not tested, it's broken

– Bruce Eckel



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.
                """

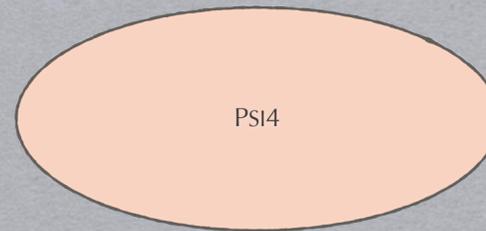
    with pytest.raises(qcelemental.ValidationError) as e:
        qcelemental.molparse.from_string(subject)

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

XXX
XXX



MACPSINET



PS14

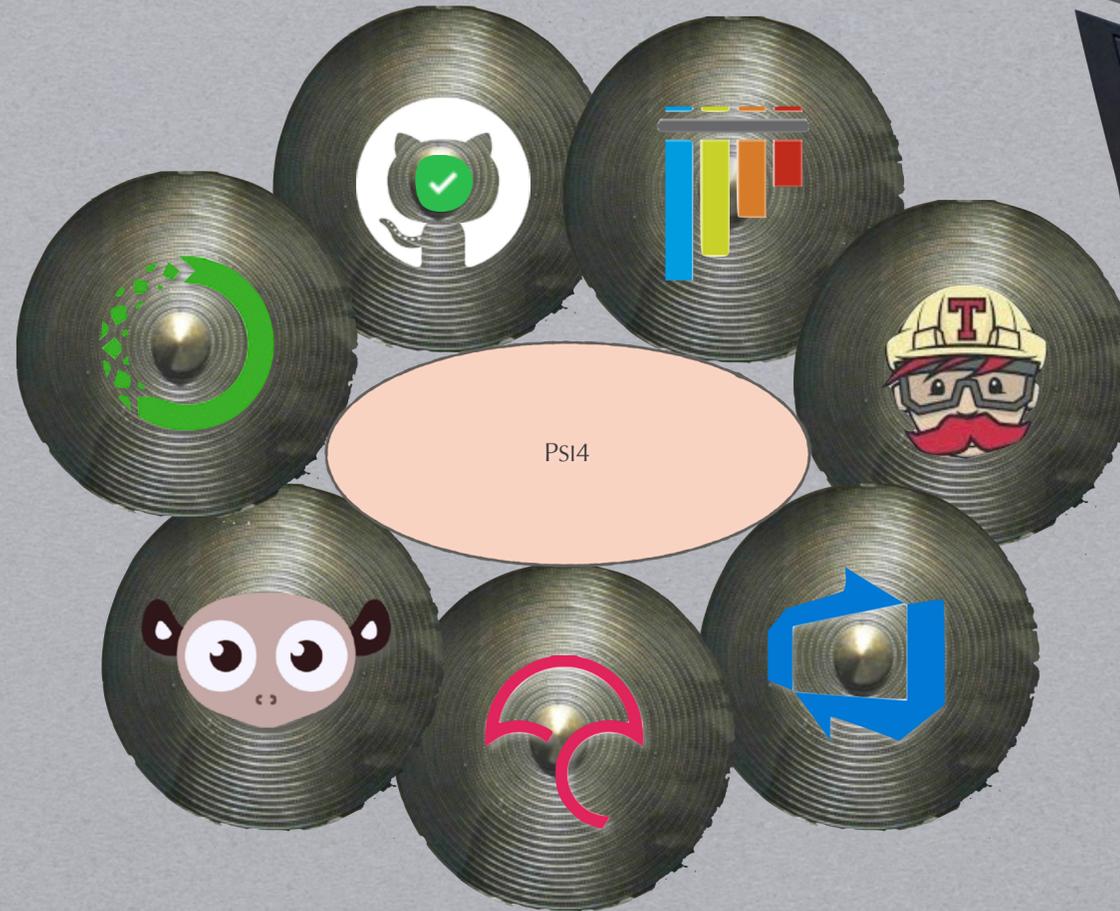


PSINET (AKA. SPINET)

XXX
XXX



MACPSINET



PSINET (AKA. SPINET)

UPSTREAM

BLAS-LAPACK

Python

Boost
PSI4
Libint

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**

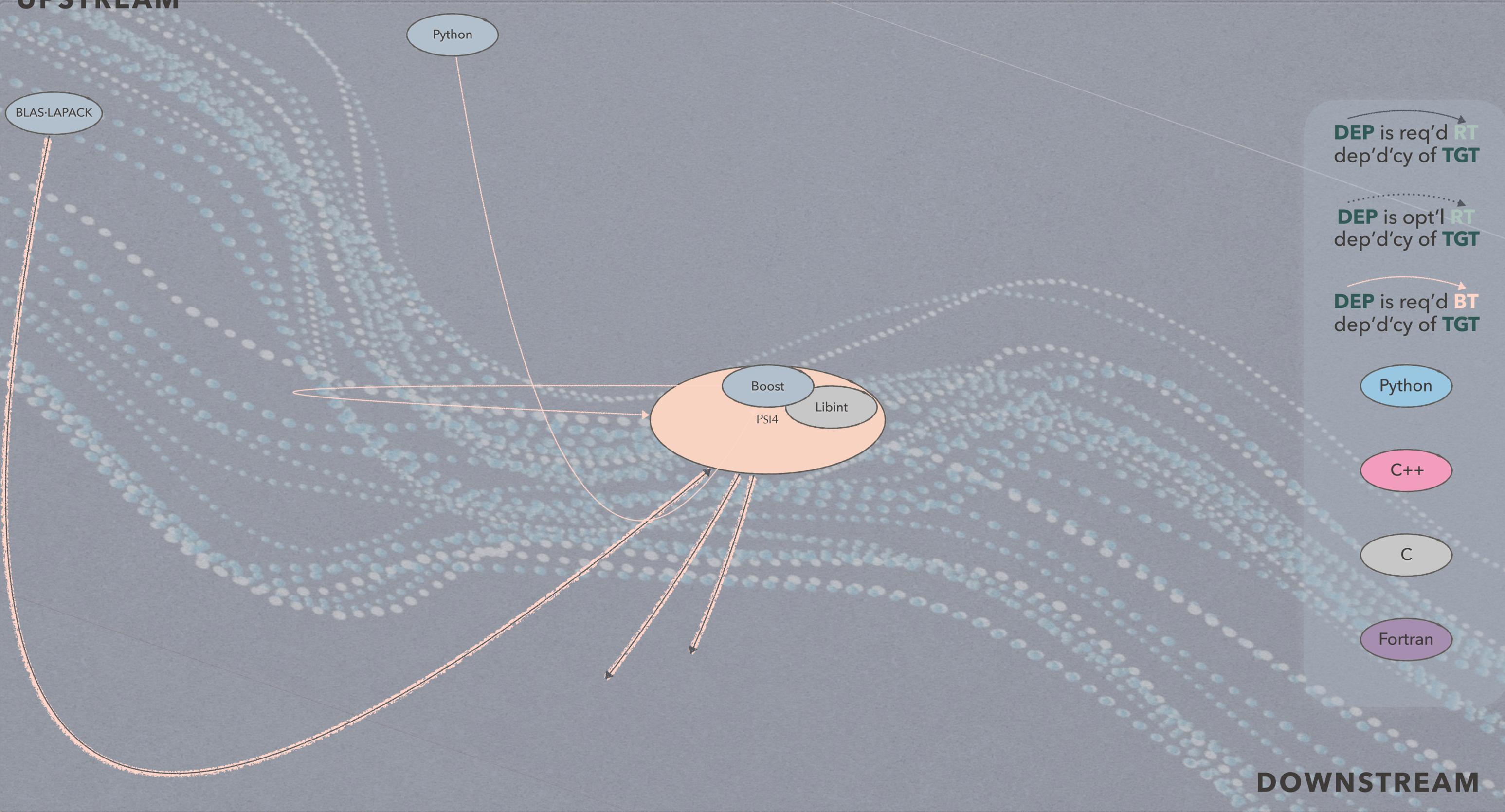
Python

C++

C

Fortran

DOWNSTREAM



UPSTREAM

BLAS-LAPACK

Python

Boost
PSI4
Libint

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**

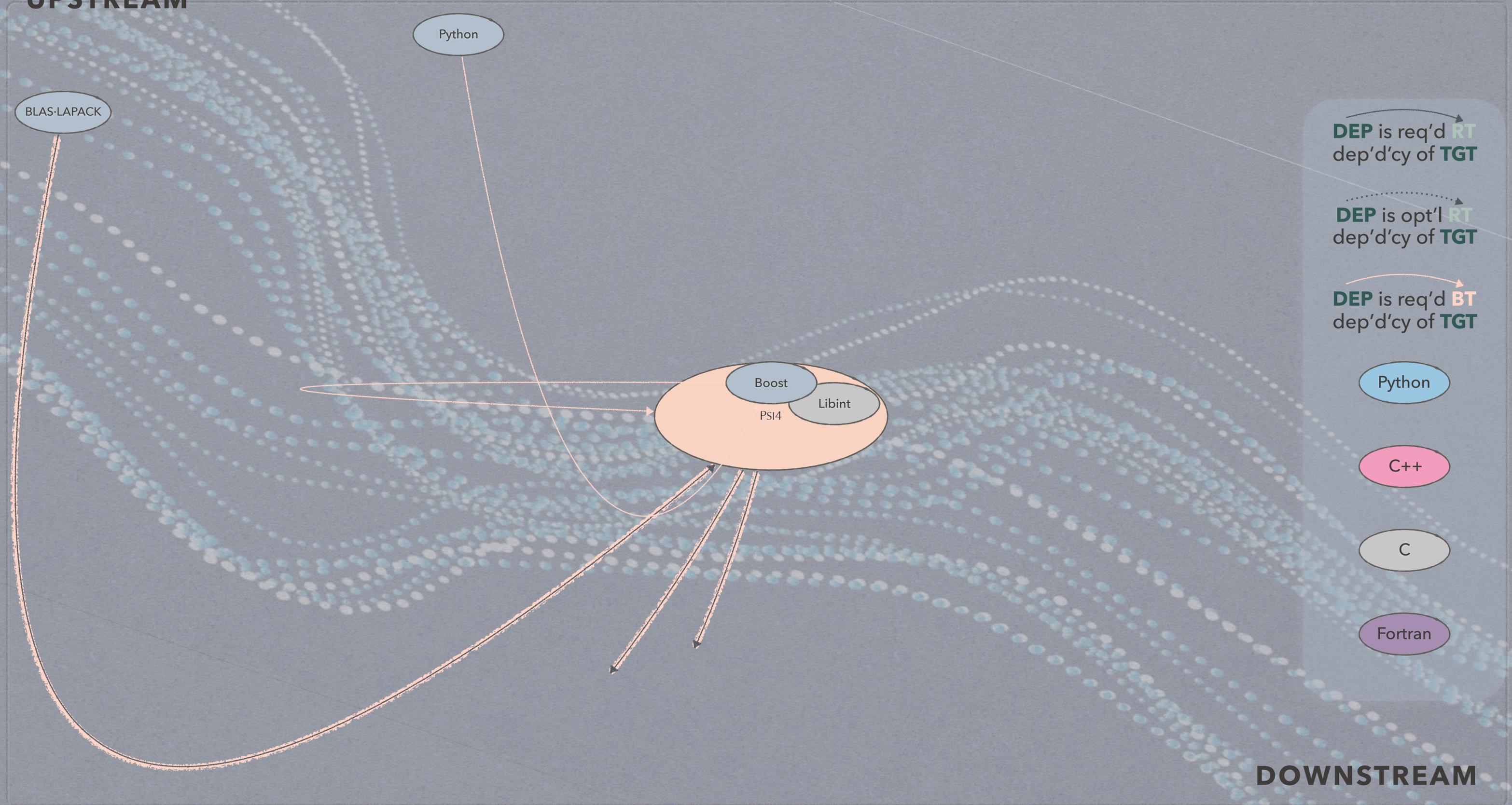
Python

C++

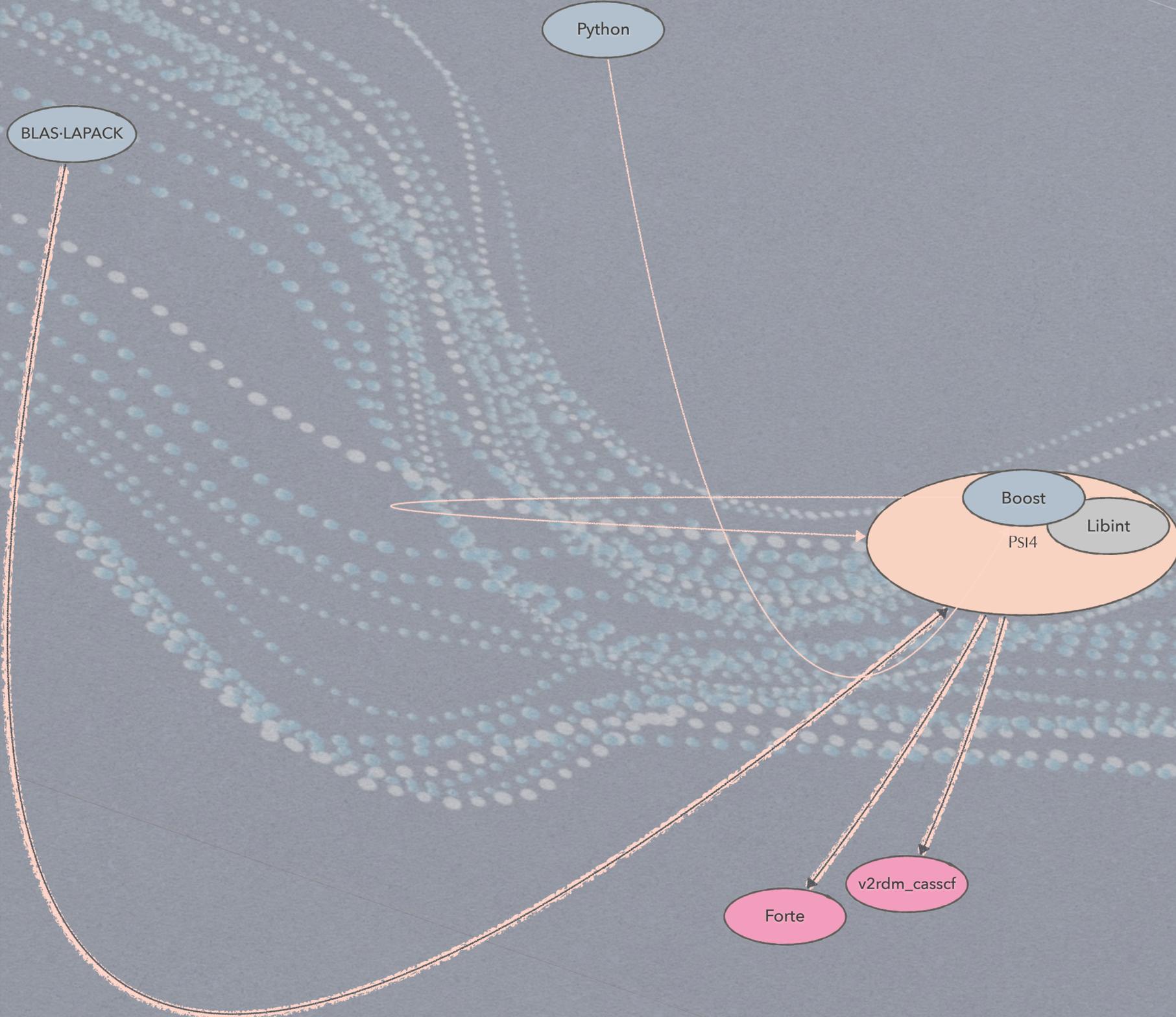
C

Fortran

DOWNSTREAM



UPSTREAM



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

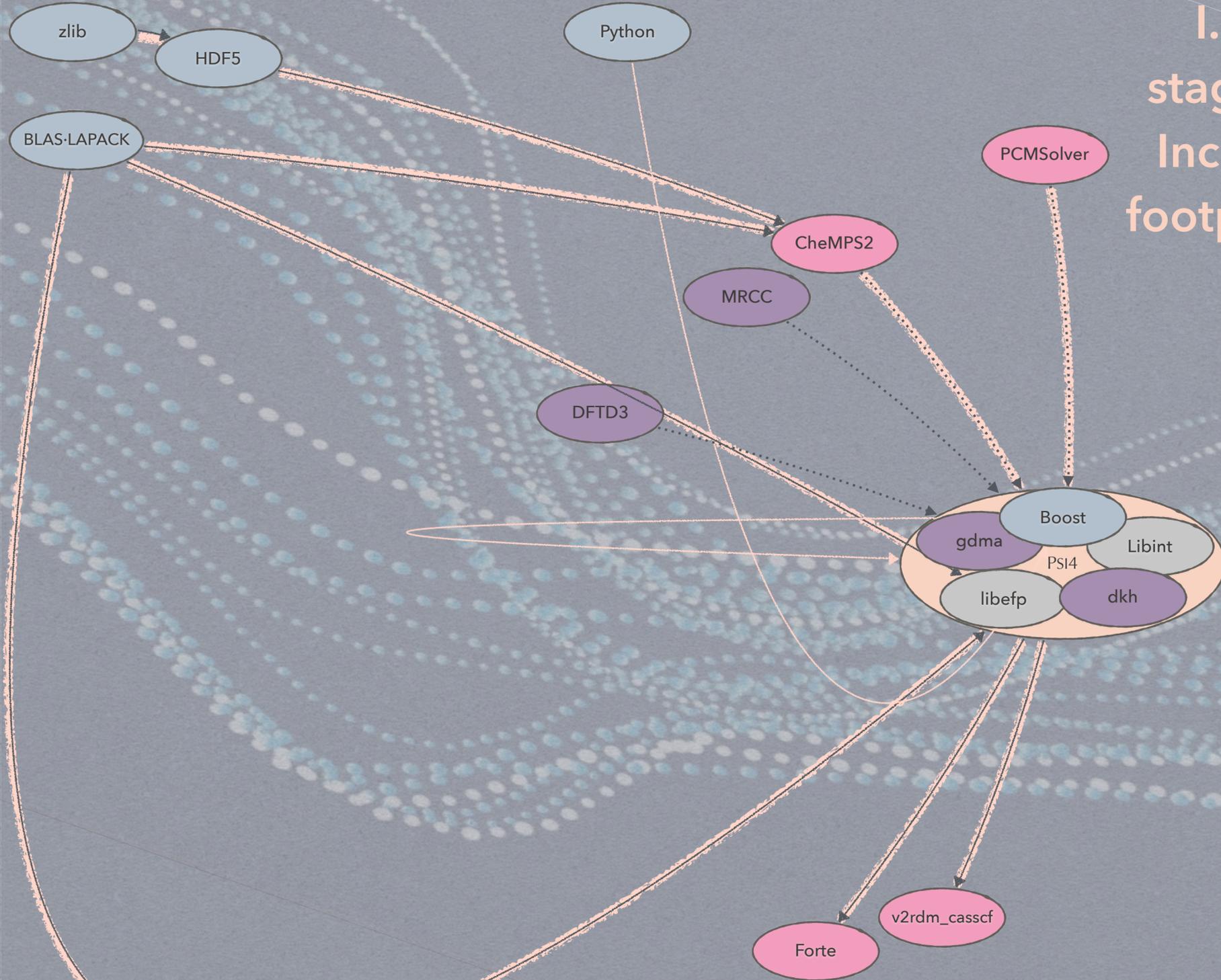
C++

C

Fortran

DOWNSTREAM

UPSTREAM



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

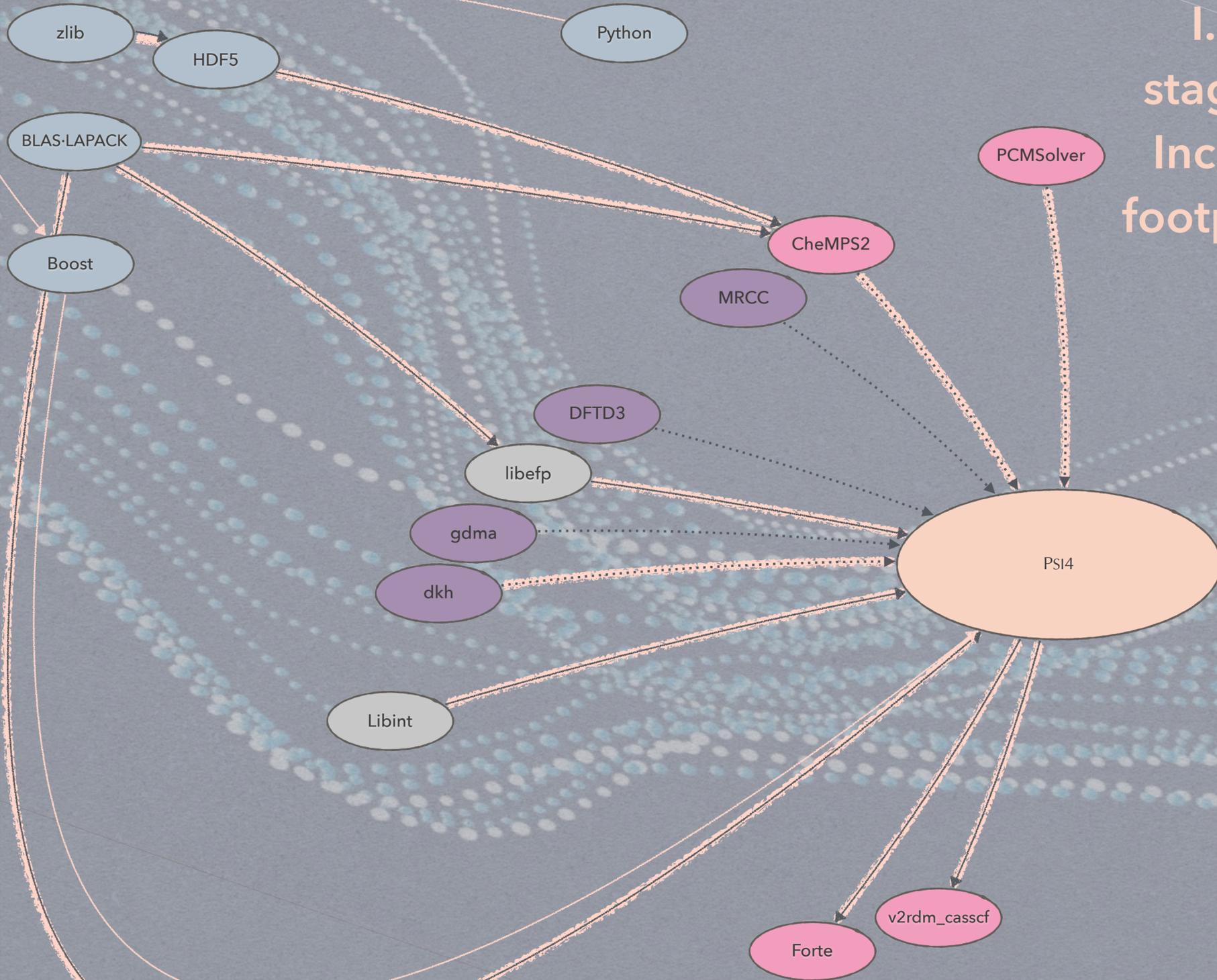
C++

C

Fortran

DOWNSTREAM

UPSTREAM



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

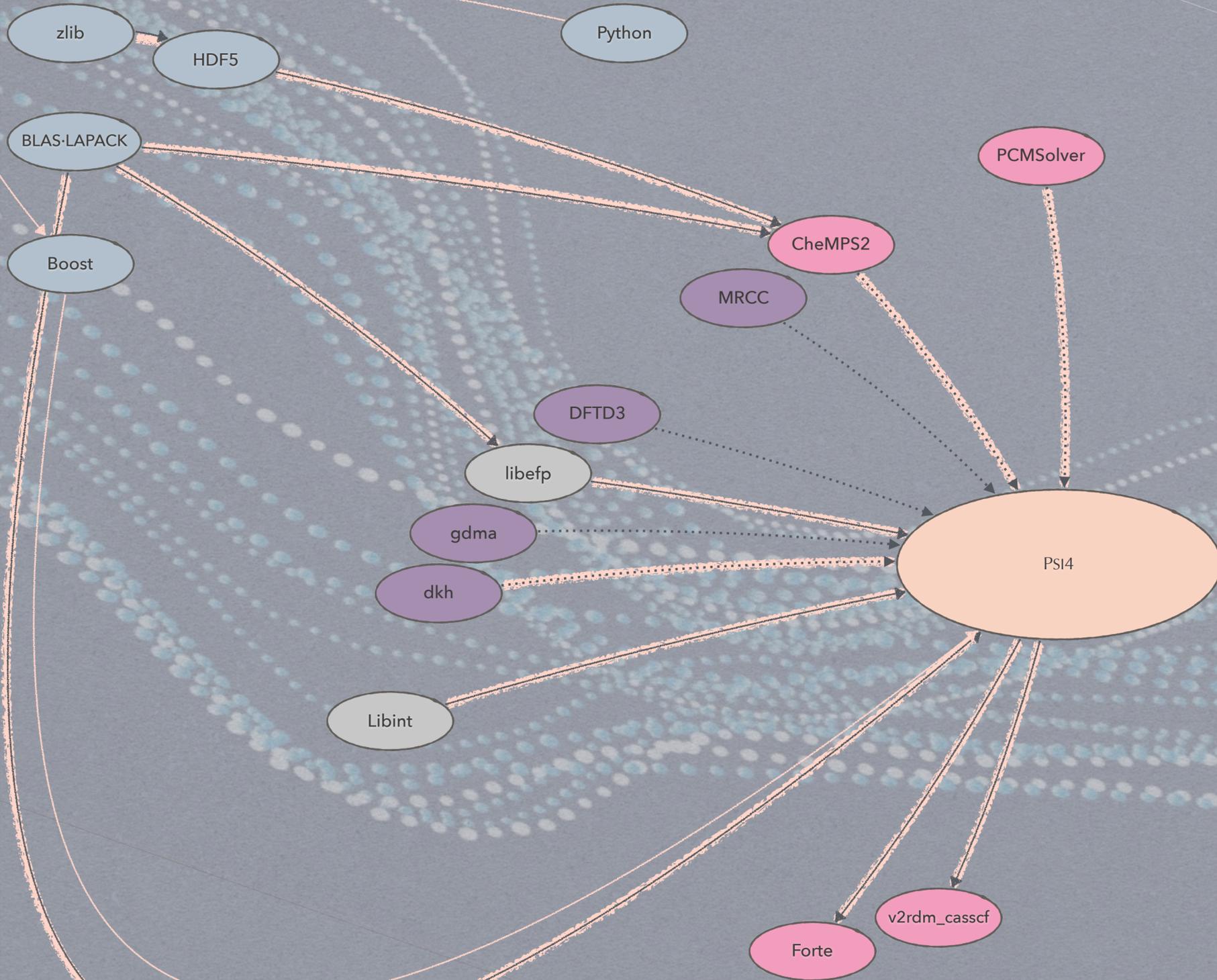
C++

C

Fortran

DOWNSTREAM

UPSTREAM



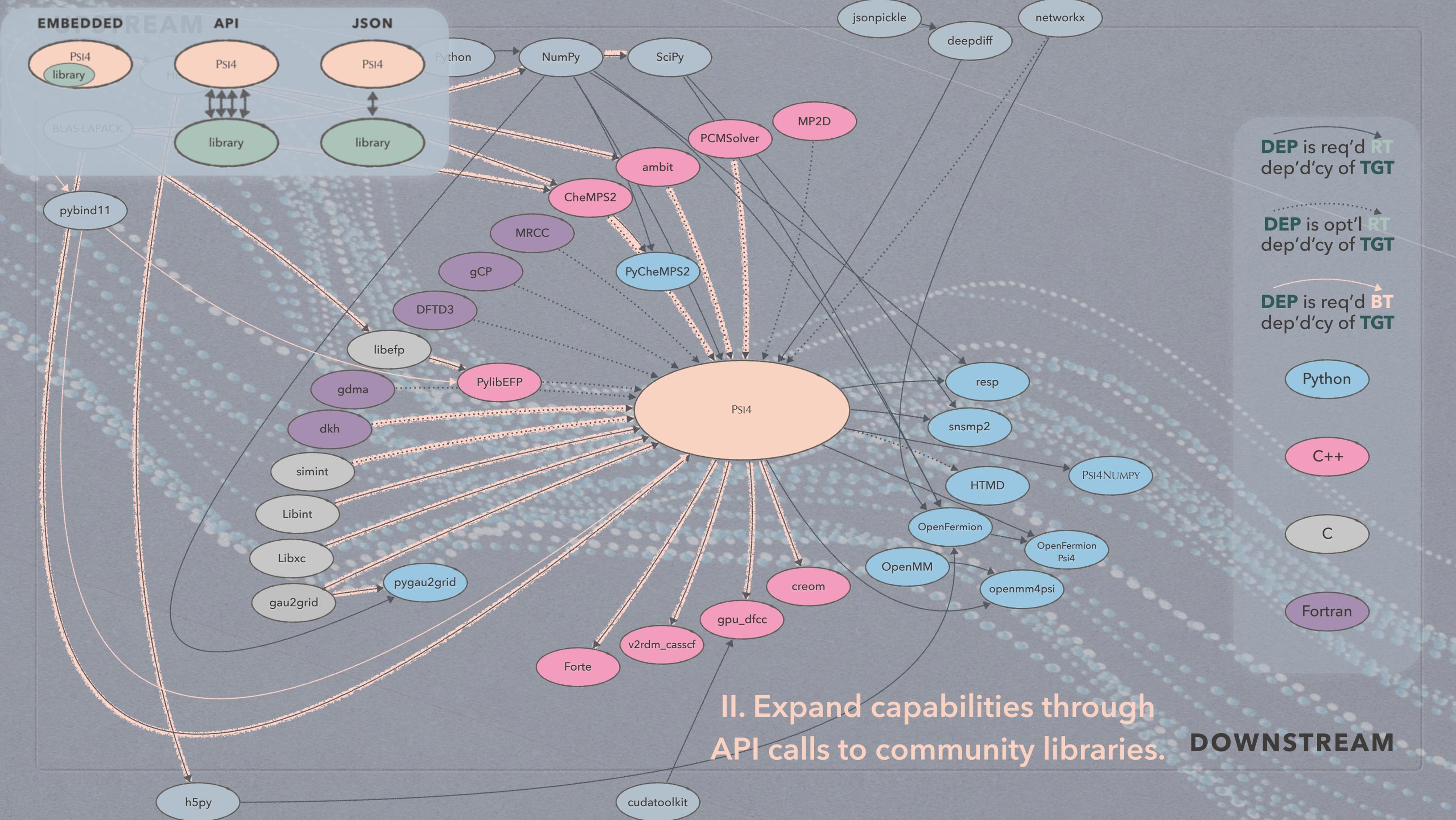
DEP is req'd **RT**
dep'd'cy of **TGT**

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**



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

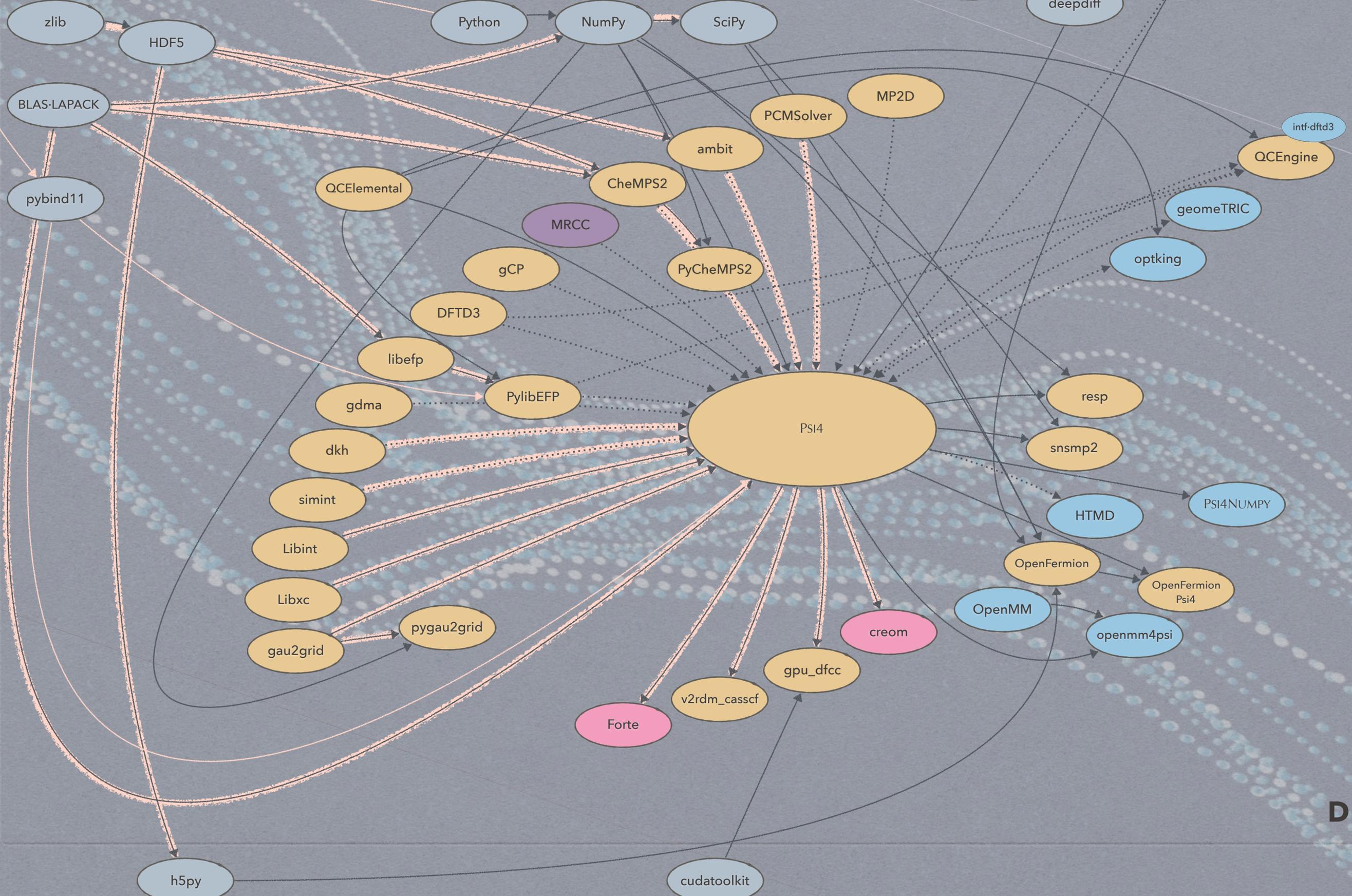
C++

C

Fortran

II. Expand capabilities through API calls to community libraries. **DOWNSTREAM**

UPSTREAM



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

UPSTREAM

CMake Helper

bin/psi4-path-advisor

Build Tools

cmake, compilers, pybind11,
MKL-devel

Req'd Py-link non-QC

Python, NumPy, networkx
deepdiff, pint, pytest

Optional Py-link QC

libefp<PylibEFP, DFTD3, gCP

Required C-link non-QC

MKL, OpenMP, HDF5

Optional C-link QC

ambit, CheMPS2, dkh, gdma,
PCMSolver, simint

Required Py-link QC

QCElemental

PSI4

Optional Py-link Psi4

OpenFermion<OF-Psi4,
resp, snsmp2

Required C-link QC

gau2grid, Libint, Libxc

Package Manager

conda

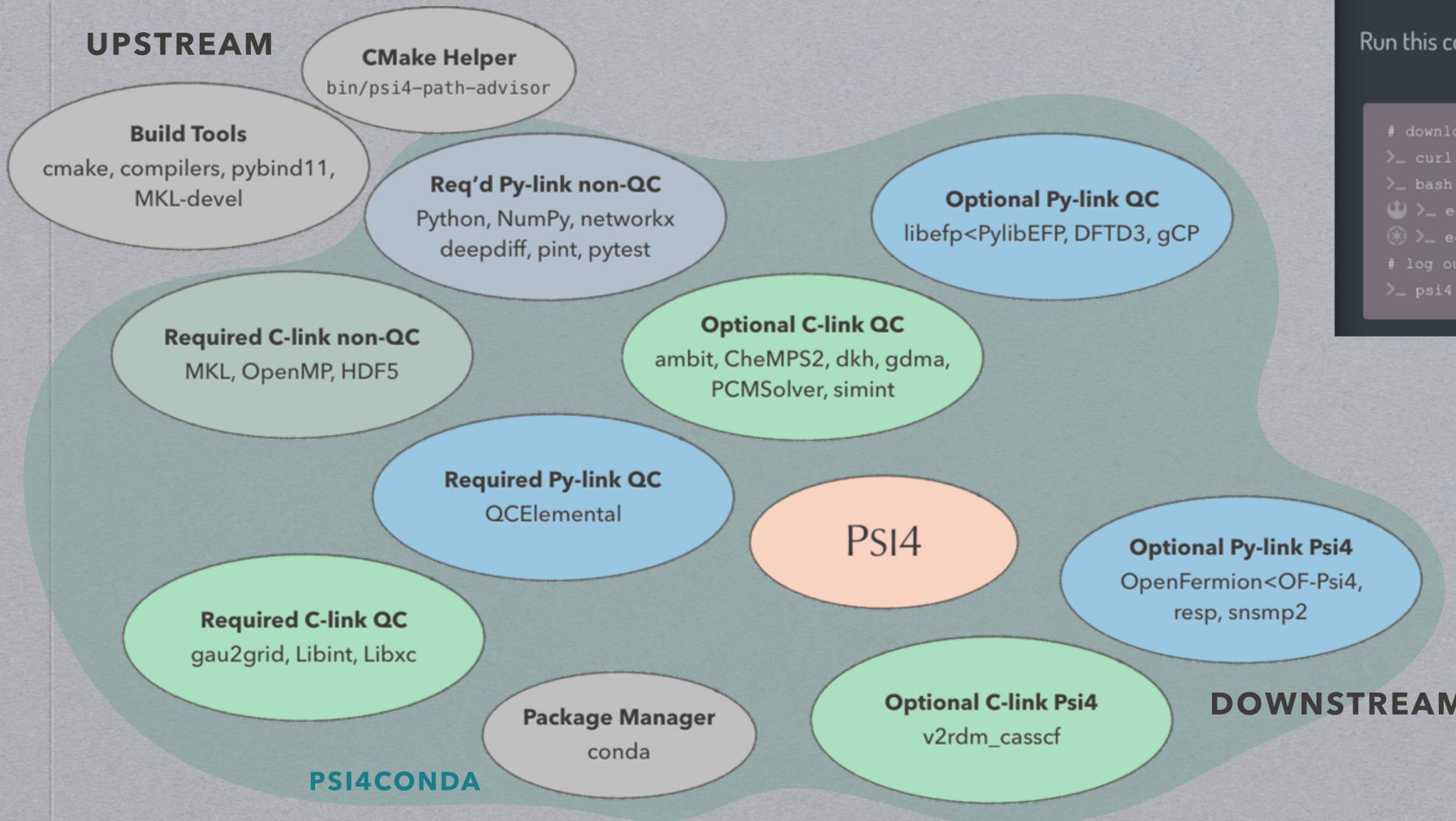
Optional C-link Psi4

v2rdm_casscf

DOWNSTREAM

CONDA FOR USERS & DEVELOPERS

UPSTREAM



Get Started with Psi4

Select Preferences

LINUX
 MACOS
 WINDOWS WSL

INSTALLER
 CONDA
 SOURCE

2.7
 3.5
 3.6
 3.7

PREV RELEASE, v1.2.1
 STABLE RELEASE, v1.3
 NIGHTLY BUILD

⚠ 64-bit; glibc 2.12 or higher. 🍏 64-bit; OS X 10.9 or higher. 🖥 64-bit; Windows Subsystem for Linux. 📄 Download standalone command-line installer. ● Use conda package manager. ⚡ Build from source using tools and dependencies from conda. 🐍 Python included, so choose the version you want, regardless of any you have.

Run this command

[DOWNLOAD PSI4CONDA INSTALLER](#)

```

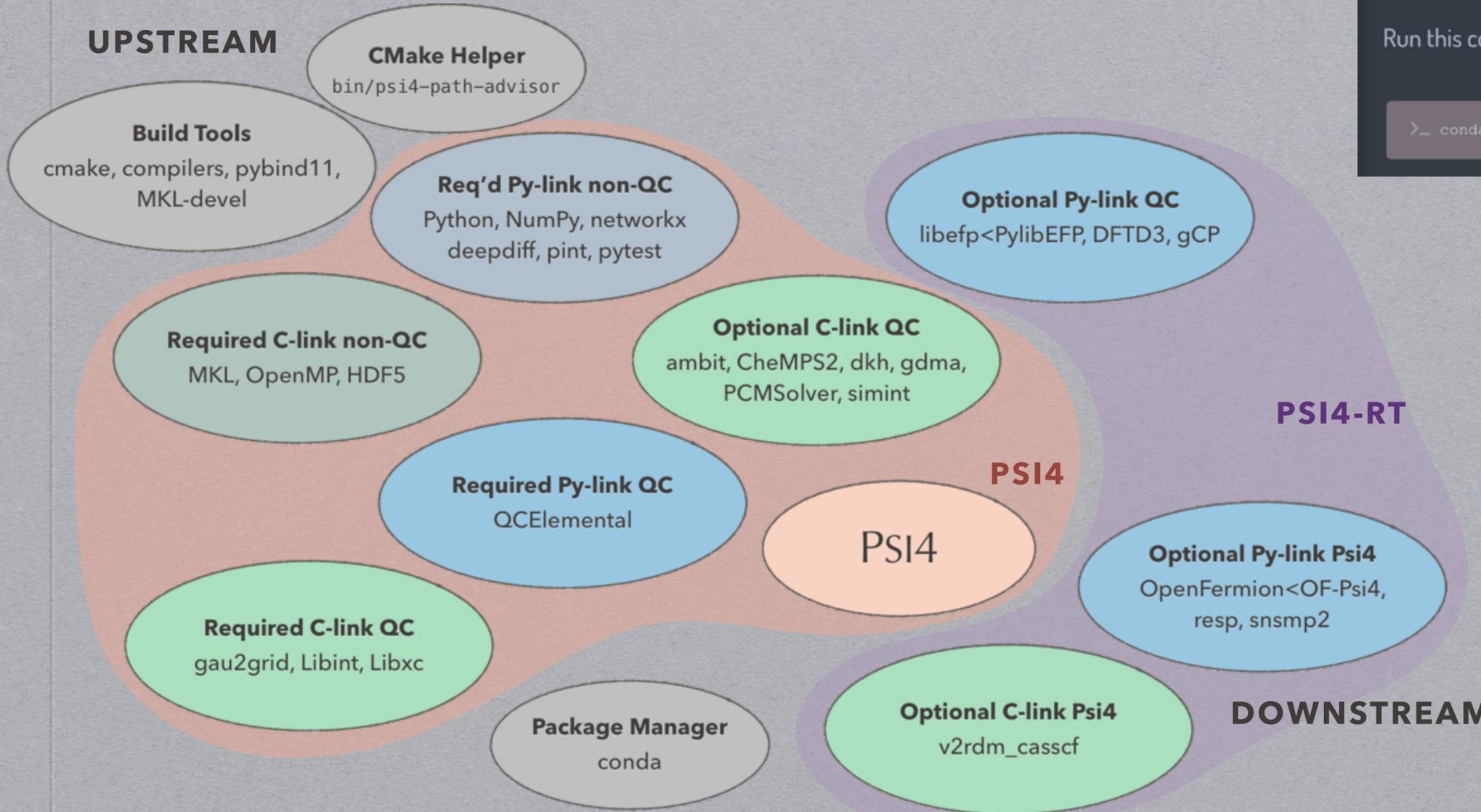
# download via button above -OR- following line
>_ curl "http://vergil.chemistry.gatech.edu/psicode-download/Psi4conda-1.3-py37-Linux-x86_64.sh" -o
>_ bash Psi4conda-1.3-py37-Linux-x86_64.sh -b -p $HOME/psi4conda
📄 >_ echo '$'. $HOME/psi4conda/etc/profile.d/conda.sh\nconda activate' >> ~/.bashrc
🌟 >_ echo "source $HOME/psi4conda/etc/profile.d/conda.csh\nconda activate" >> ~/.tcshrc
# log out, log back in so conda and psi4 in path
>_ psi4 --test
    
```

PY-WARY USERS: PSI4CONDA
CONDA INSTALLER

PSI4 · dependencies · add-ons

CONDA FOR USERS & DEVELOPERS

UPSTREAM



Get Started with PSI4

Select Preferences

LINUX
 MACOS
 WINDOWS WSL

INSTALLER
 CONDA
 SOURCE

2.7
 3.5
 3.6
 3.7

PREV RELEASE, v1.2.1
 STABLE RELEASE, v1.3
 NIGHTLY BUILD

⚠ 64-bit; glibc 2.12 or higher. 🍏 64-bit; OS X 10.9 or higher. 🖥 64-bit; Windows Subsystem for Linux. 📄 Download standalone command-line installer. ● Use conda package manager. </> Build from source using tools and dependencies from conda. 🐍 Python included, so choose the version you want, regardless of any you have.

Run this command

[GOTO MINICONDA INSTALLERS](#)

```
>_ conda install psi4 psi4-rt python=3.7 -c psi4
```

PY-FRIENDLY USERS: CONDA PACKAGE

PSI4 · dependencies · add-ons

PY-WARY USERS: CONDA INSTALLER

PSI4 · dependencies · add-ons

PSI4CONDA

57K

CONDA FOR USERS & DEVELOPERS

UPSTREAM

CMake Helper
bin/psi4-path-advisor **PSI4-DEV**

Build Tools
cmake, compilers, pybind11, MKL-devel

Req'd Py-link non-QC
Python, NumPy, networkx, deepdiff, pint, pytest

Optional Py-link QC
libefp<PylibEFP, DFTD3, gCP

Required C-link non-QC
MKL, OpenMP, HDF5

Optional C-link QC
ambit, CheMPS2, dkh, gdma, PCMSolver, simint

Required Py-link QC
QCElemental

PSI4

Optional Py-link Psi4
OpenFermion<OF-Psi4, resp, snsmp2

Required C-link QC
gau2grid, Libint, Libxc

Package Manager
conda

Optional C-link Psi4
v2rdm_casscf

DOWNSTREAM

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 57K
PSI4 · dependencies · add-ons

Get Started with Psi4

Select Preferences

LINUX
 MACOS
 WINDOWS WSL

INSTALLER
 CONDA
 SOURCE

2.7
 3.5
 3.6
 3.7

PREV RELEASE, v1.2.1
 STABLE RELEASE, v1.3
 NIGHTLY BUILD

⚠ 64-bit; glibc 2.12 or higher. 🍏 64-bit; OS X 10.9 or higher. 🖥 64-bit; Windows Subsystem for Linux. 📄 Download standalone command-line installer. ● Use conda package manager. ⚡ Build from source using tools and dependencies from conda. 🐍 Python included, so choose the version you want, regardless of any you have.

Run this command

[GOTO MINICONDA INSTALLERS](#)

```

>_ git clone https://github.com/psi4/psi4.git && cd psi4
>_ conda create -n p4dev psi4-dev python=3.7 -c psi4/label/dev
>_ conda activate p4dev
>_ 'psi4-path-advisor --gcc'
>_ cd objdir && make -j`getconf _NPROCESSORS_ONLN`
    
```

CONDA FOR USERS & DEVELOPERS



UPSTREAM

CMake Helper
bin/psi4-path-advisor

PSI4-DEV

Build Tools
cmake, compilers, pybind11,
MKL-devel

Req'd Py-link non-QC
Python, NumPy, networkx
deepdiff, pint, pytest

Optional Py-link QC
libefp<PylibEFP, DFTD3, gCP

Required C-link non-QC
MKL, OpenMP, HDF5

Optional C-link QC
ambit, CheMPS2, dkh, gdma,
PCMSolver, simint

Required Py-link QC
QCElemental

PSI4

Optional Py-link Psi4
OpenFermion<OF-Psi4,
resp, snsmp2

Required C-link QC
gau2grid, Libint, Libxc

Package Manager
conda

Optional C-link Psi4
v2rdm_casscf

DOWNSTREAM

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

PSI4 · dependencies · add-ons

50K

PY-WARY USERS: CONDA INSTALLER

PSI4CONDA

PSI4 · dependencies · add-ons

57K

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.

Wall Time wrt 1.3 binary



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, giving user 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

MolSSI / QCSchema

Unwatch 18 Unstar 44 Fork 24

Code Issues 17 Pull requests 5 Projects 0 Wiki Insights

A Schema for Quantum Chemistry <http://molssi-qc-schema.readthedocs.i...>

153 commits 1 branch 0 releases 10 contributors BSD-3-Clause

Branch: master New pull request Create new file Upload files Find File Clone or download

loriab and dgasmith qcschema_input and qcschema_molecule updating (#60) ... Latest commit c855c2f 29 days ago

VALIDATING PYTHON OBJECTS HERE

QCElemental / qcelemental / models

Create new file Upload files Find file History

loriab typehint new models.Molecule fns. yapf Latest commit 82304b0 2 days ago

..

__init__.py	Results: Exposes property object for use elsewhere	24 days ago
common_models.py	rename to DriverEnum.derivative_int	16 days ago
molecule.py	typehint new models.Molecule fns. yapf	2 days ago
procedures.py	Models: More strict output validation of required fields	19 days ago
results.py	Models: More strict output validation of required fields	19 days ago

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, -0.06563853809917,
                              '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

IO

JSON:
QCSchema

```
{
  '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

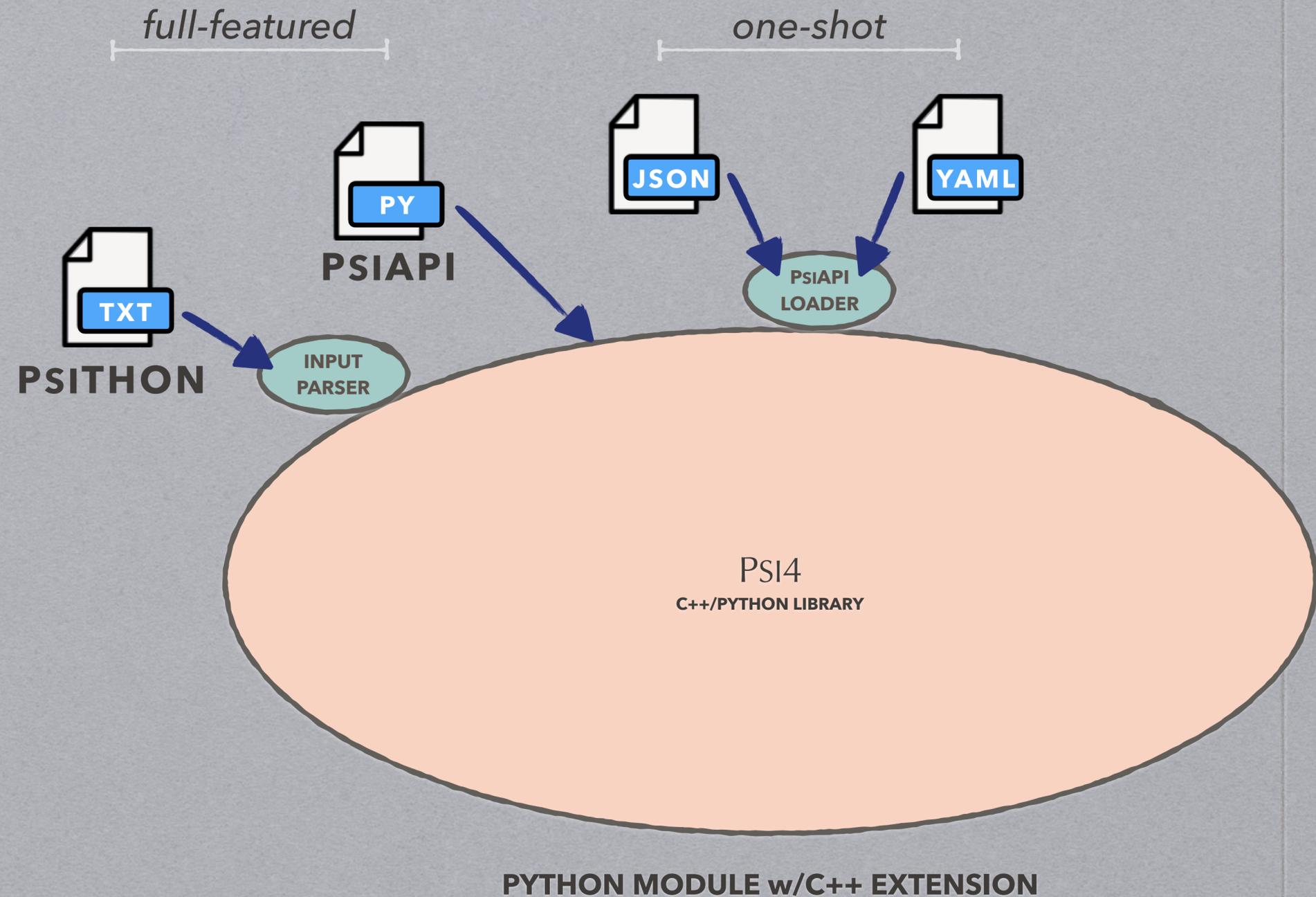
QCJOB DICT

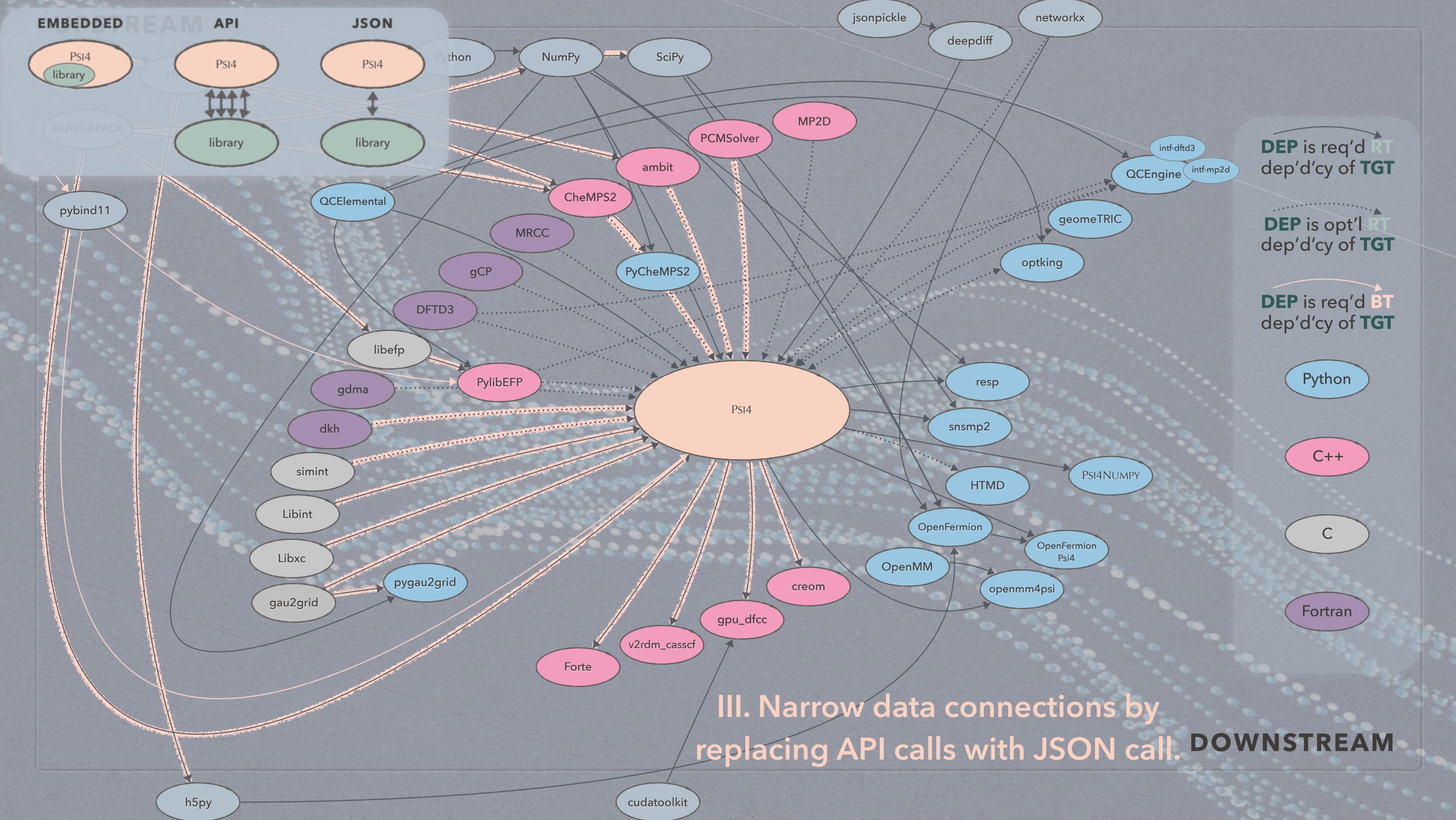
CHARACTERISTICS

- 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





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

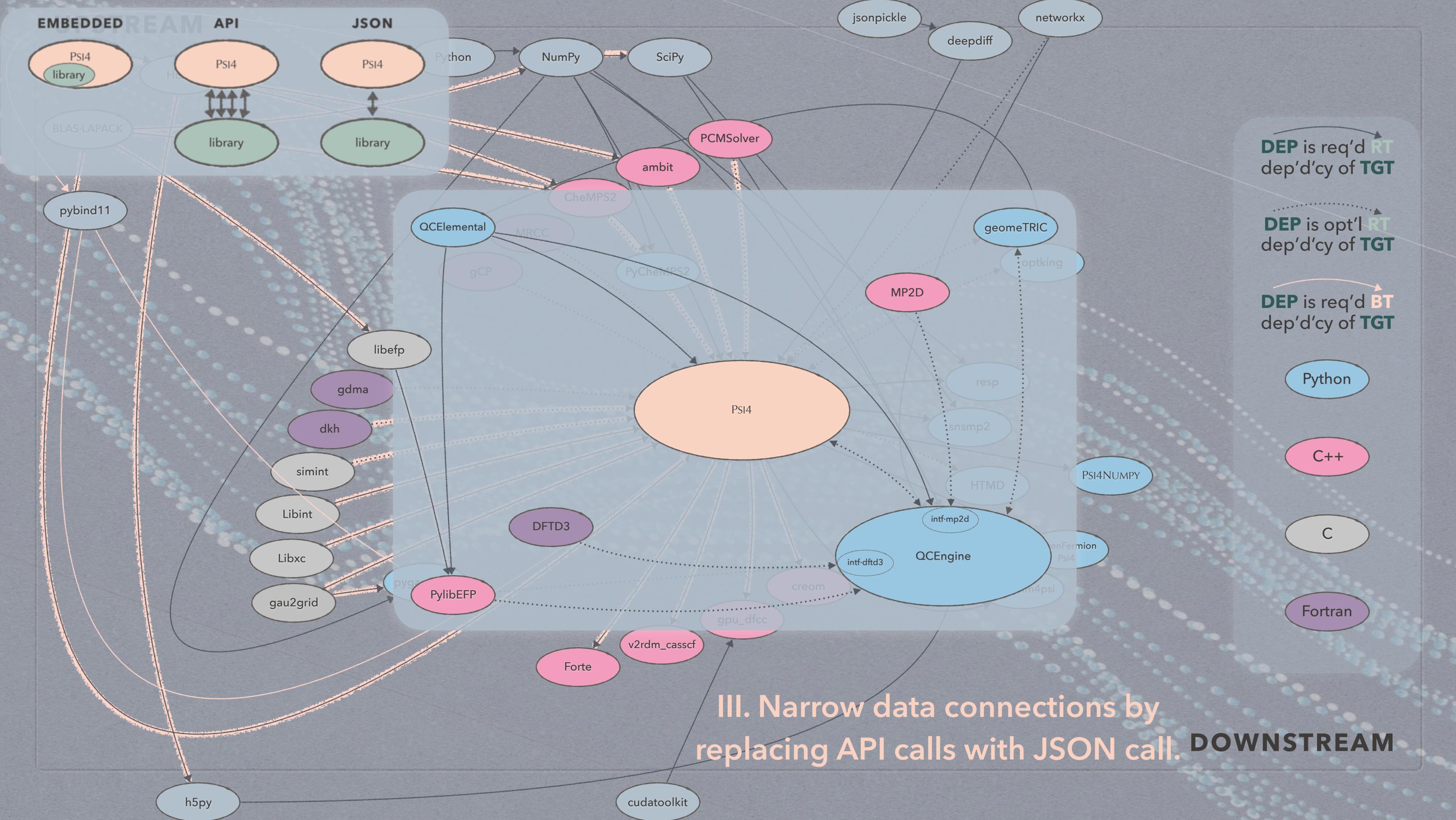
Python

C++

C

Fortran

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



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

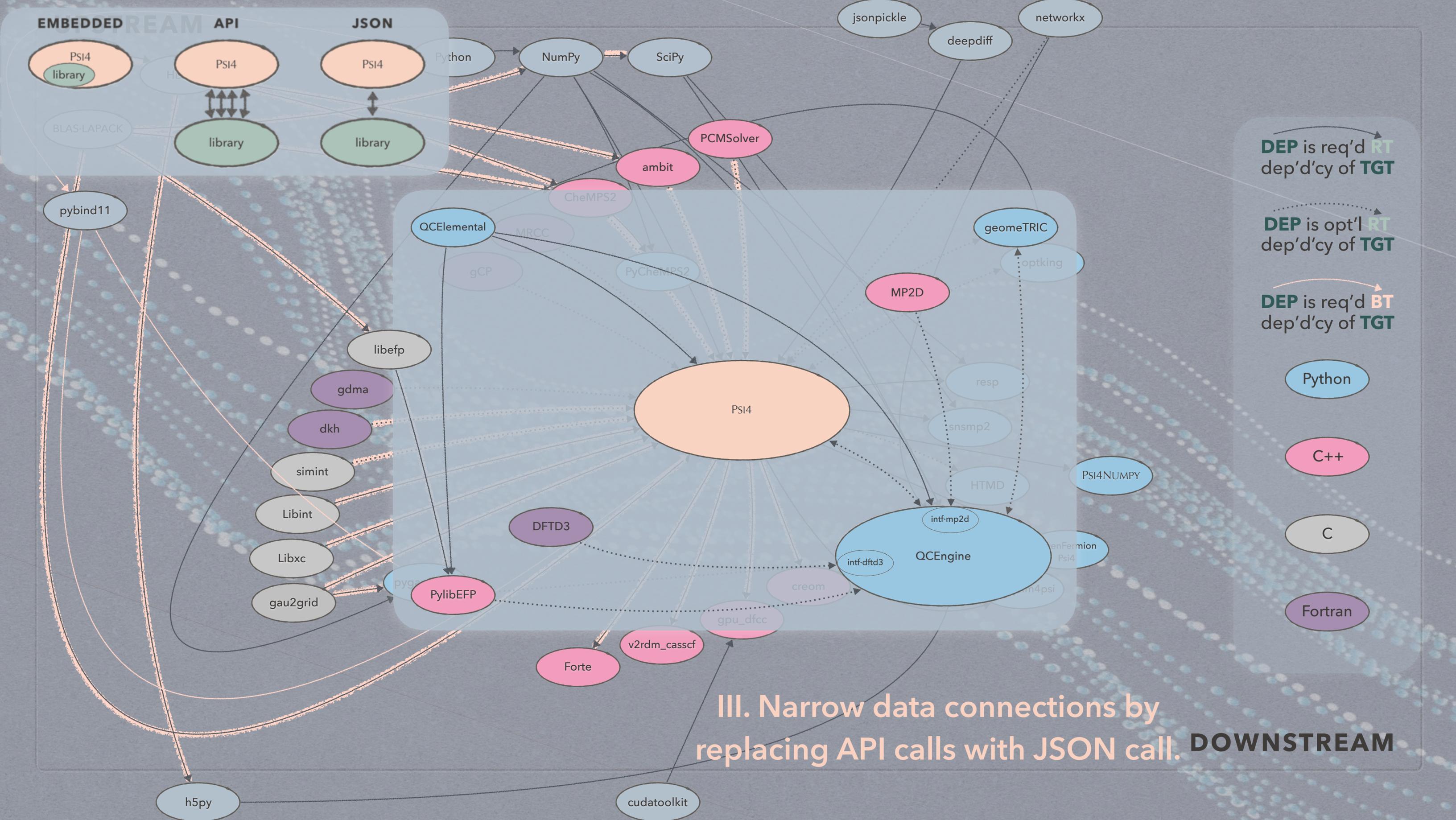
Python

C++

C

Fortran

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



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

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
```

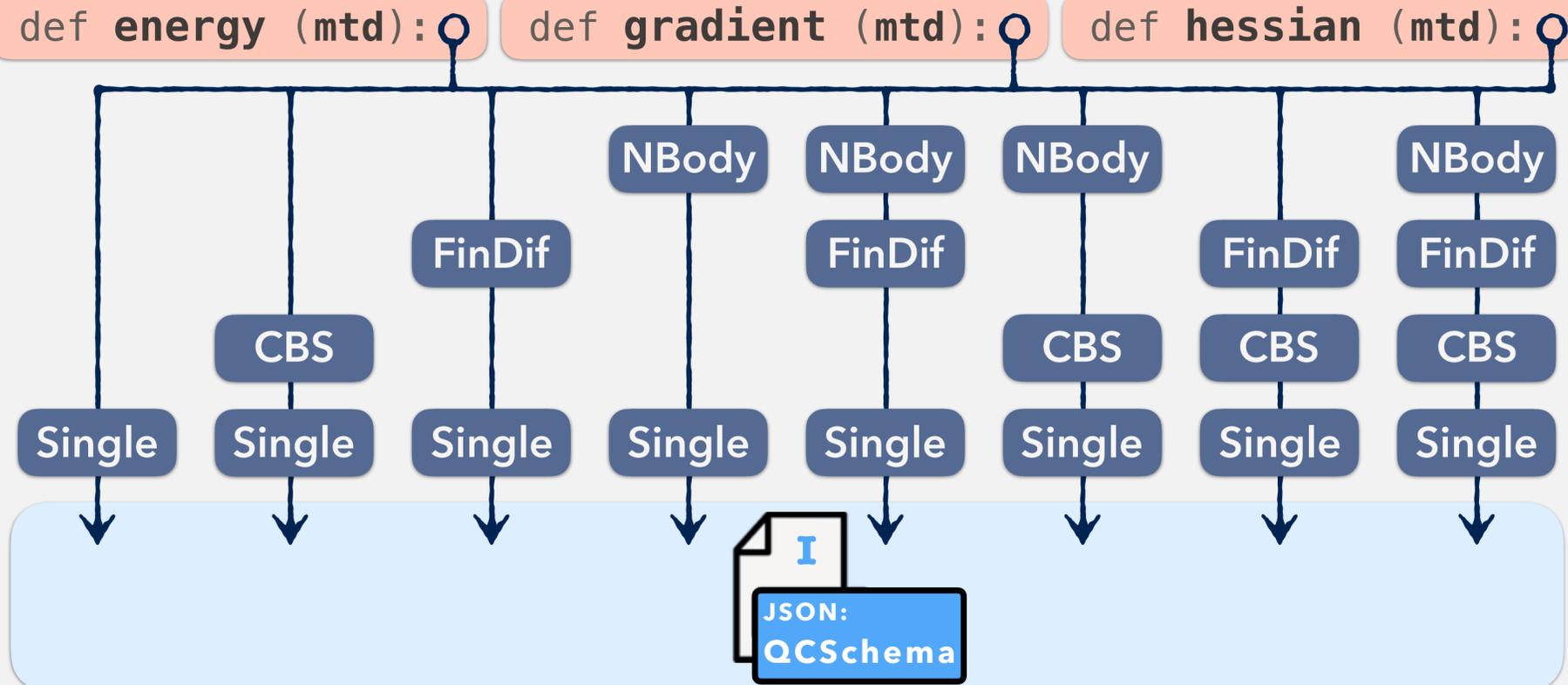
ASM Assemble derivative results from displacements.

```
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.



```
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.

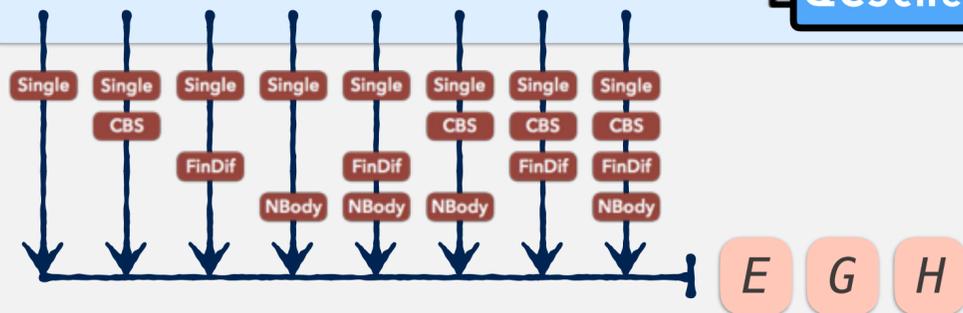
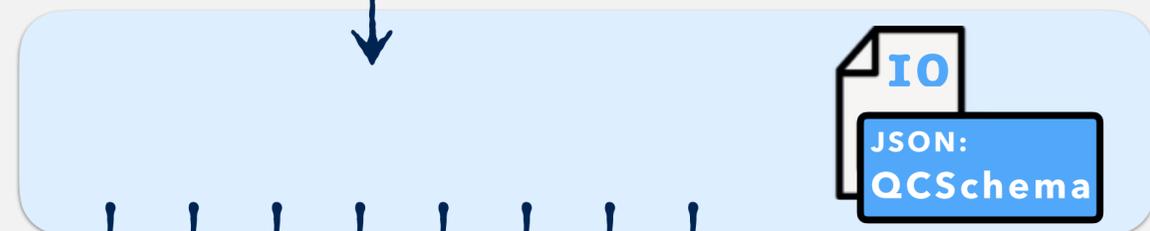
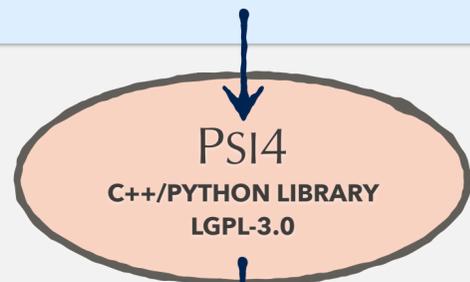
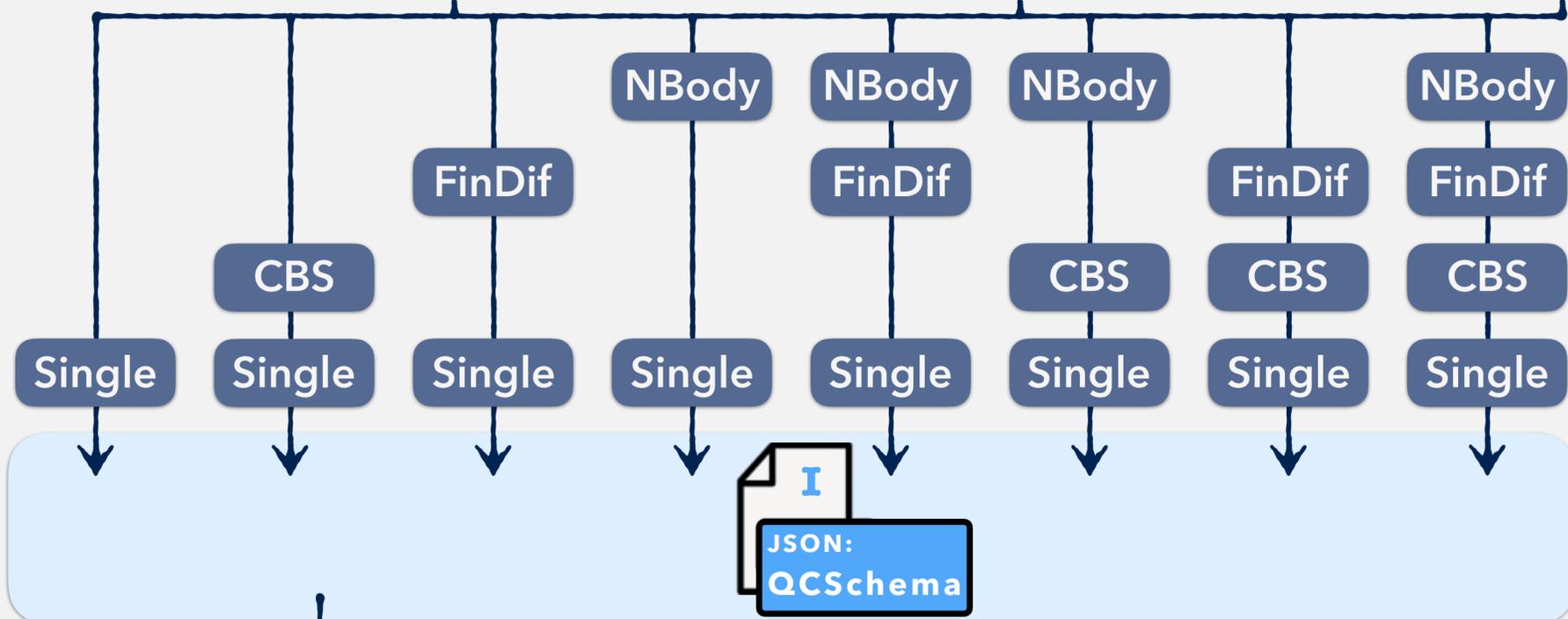
class **SingleResult** ():

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

ASM Return analytic energy, gradient, or Hessian.

DISTRIBUTED DRIVER

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.



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.

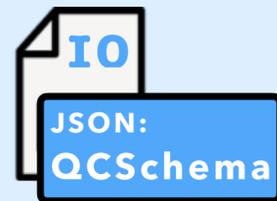
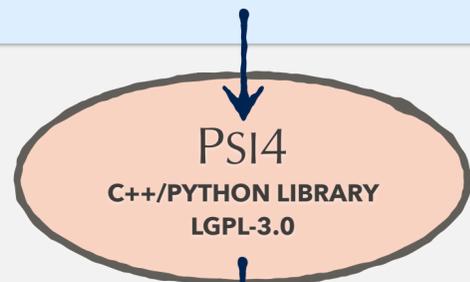
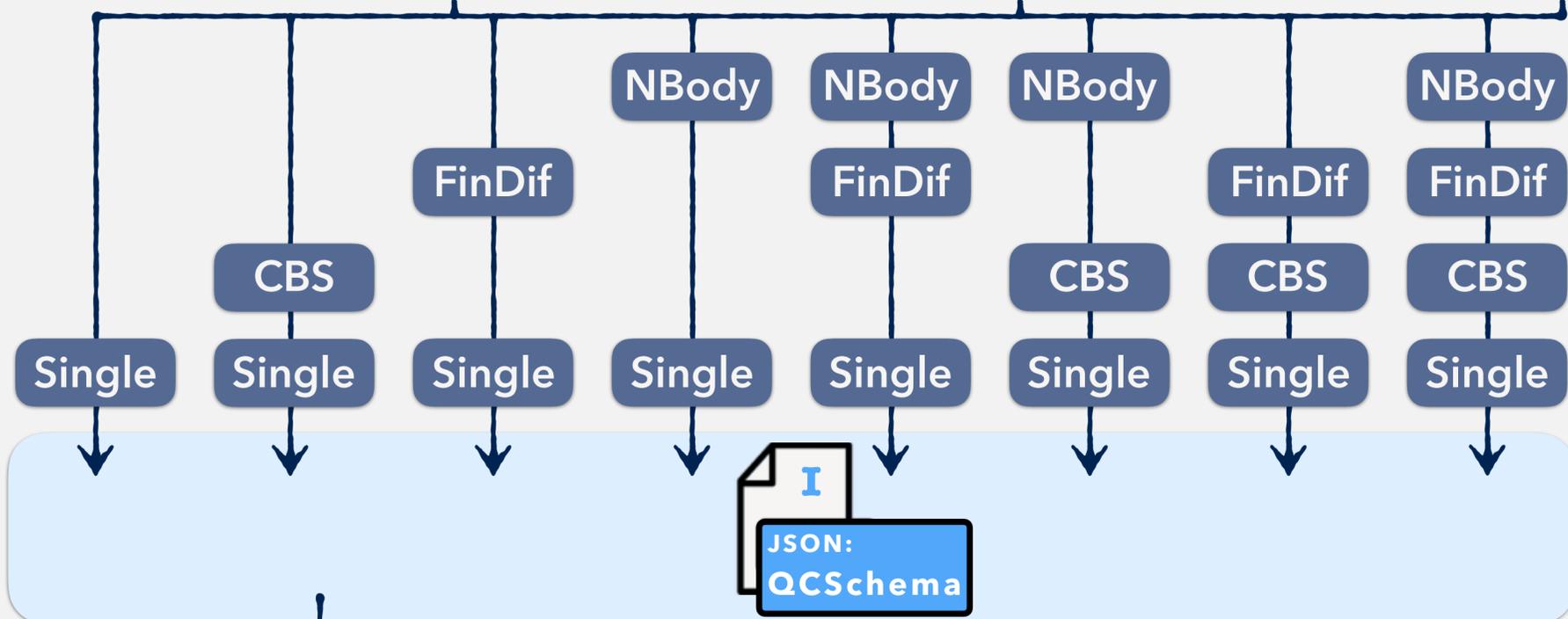
class SingleResult ():

PLAN molecule & method & func unchanged. return json

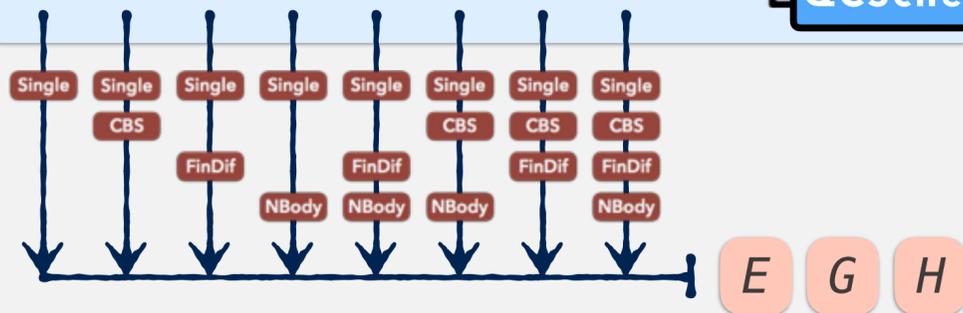
ASM Return analytic energy, gradient, or Hessian.

DISTRIBUTED DRIVER

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



● 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.



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.

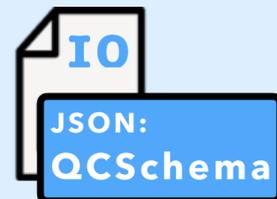
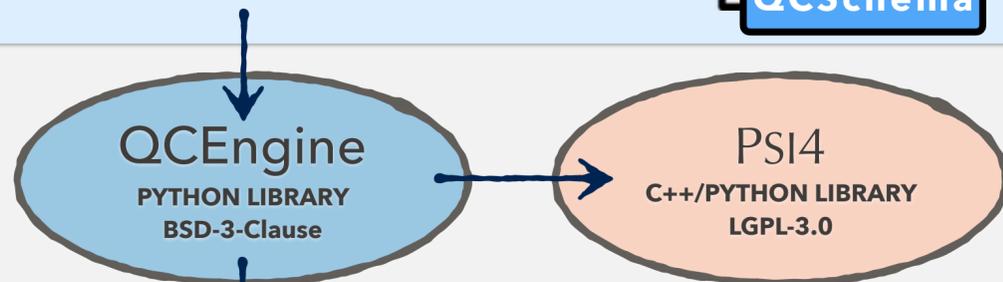
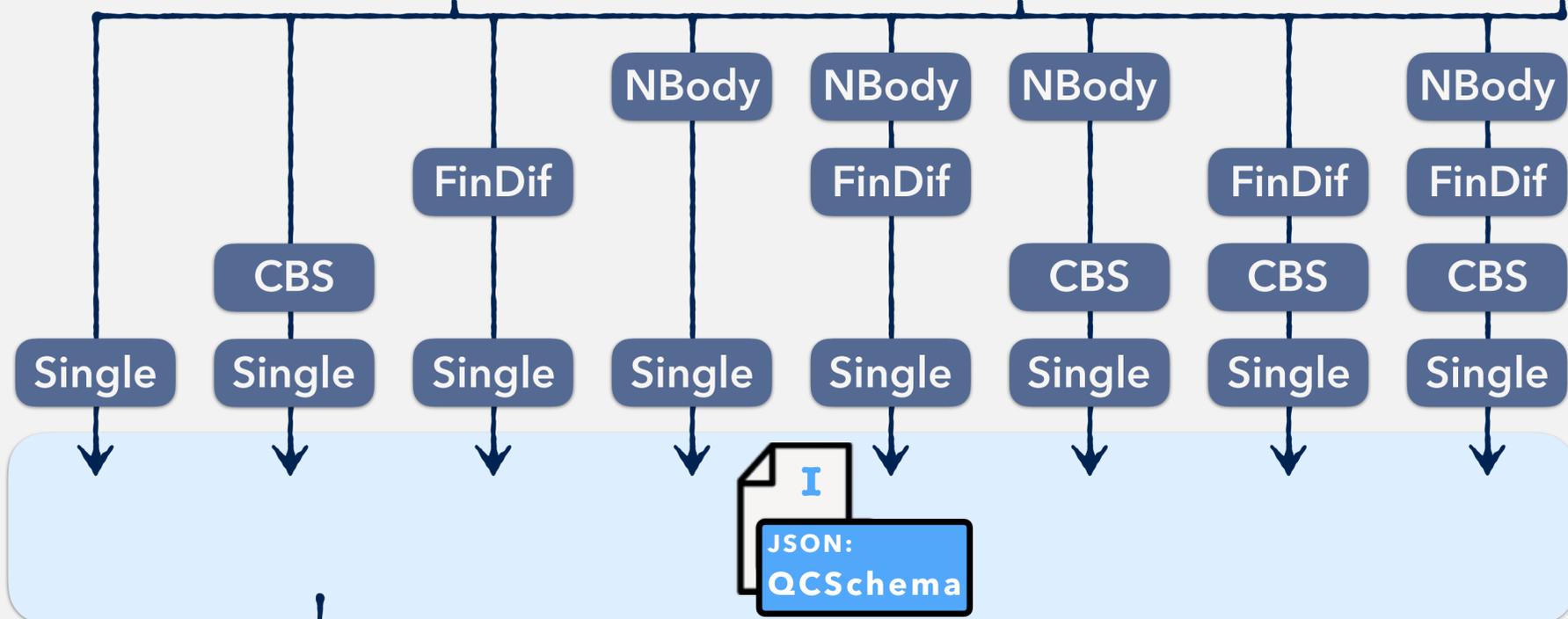
class **SingleResult** ():

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

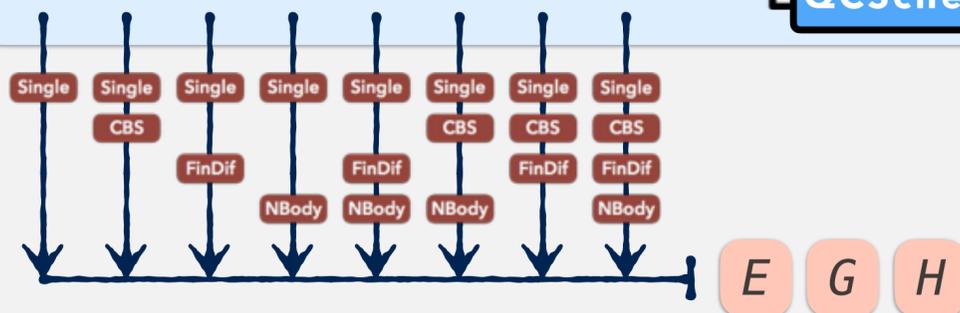
ASM Return analytic energy, gradient, or Hessian.

DISTRIBUTED DRIVER

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



● 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.



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.

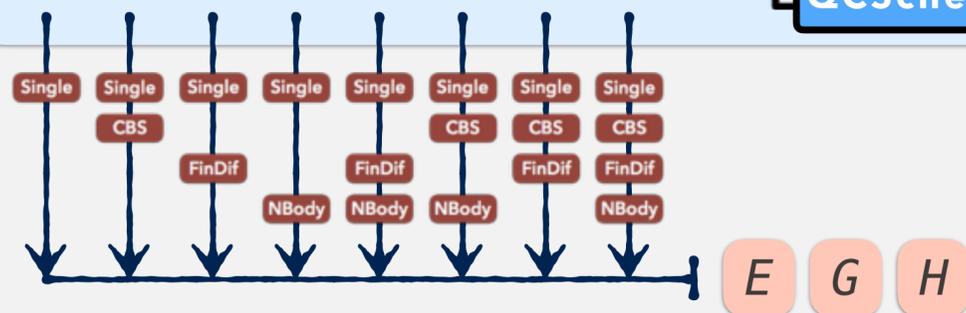
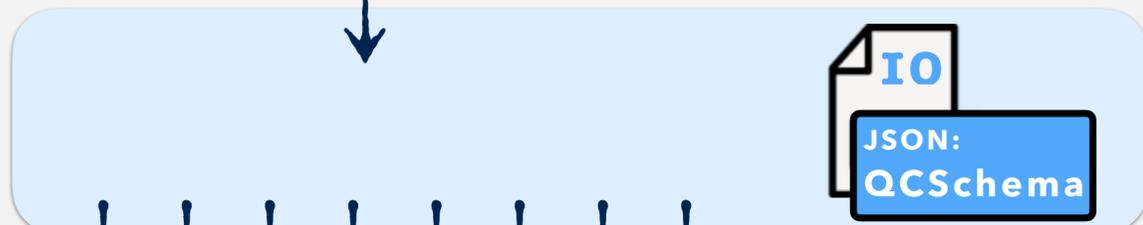
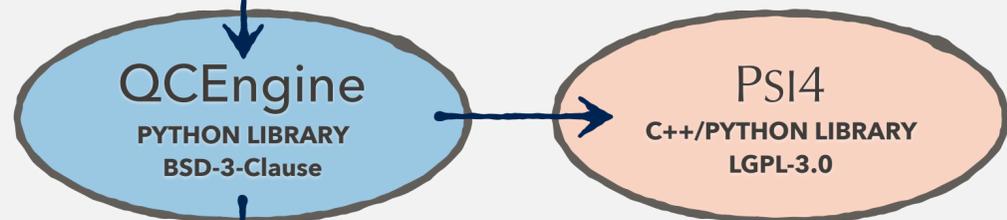
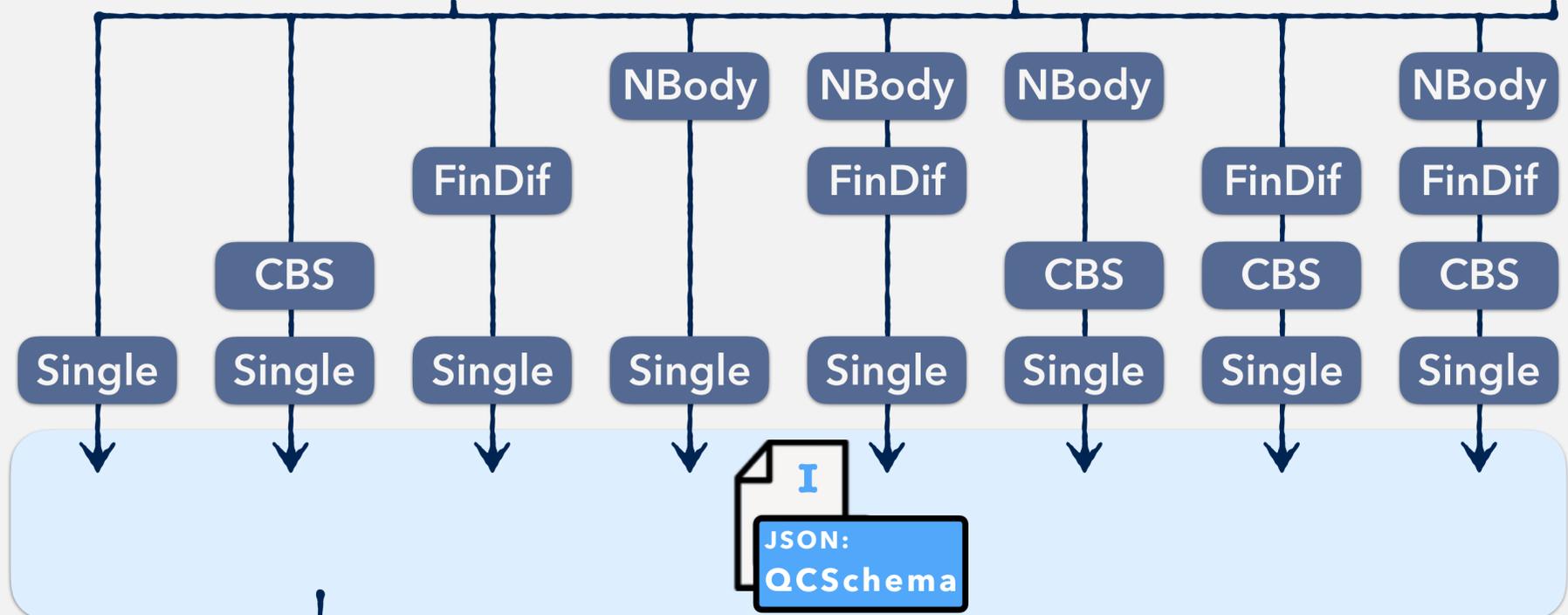
class **SingleResult** ():

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

ASM Return analytic energy, gradient, or Hessian.

DISTRIBUTED DRIVER

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



- 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.



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.

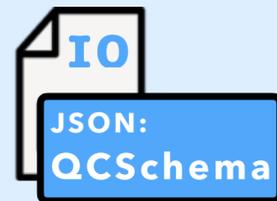
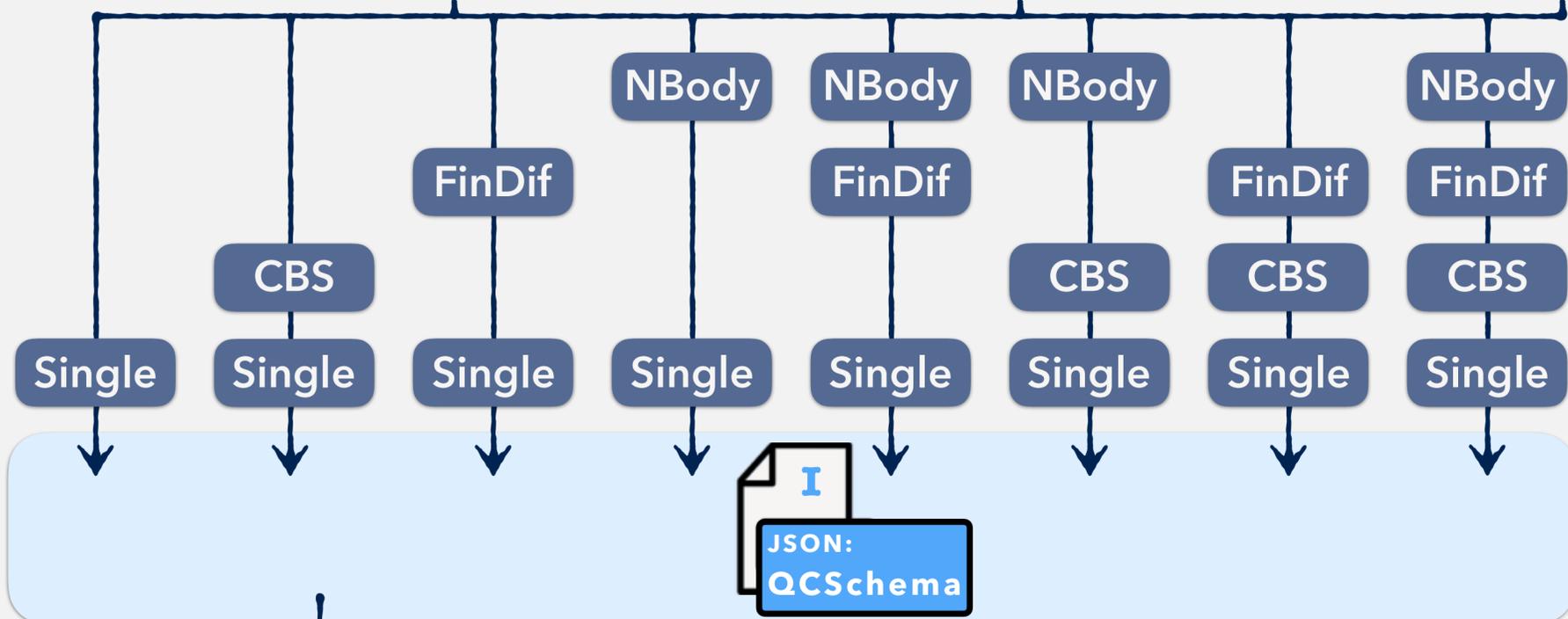
class **SingleResult** ():

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

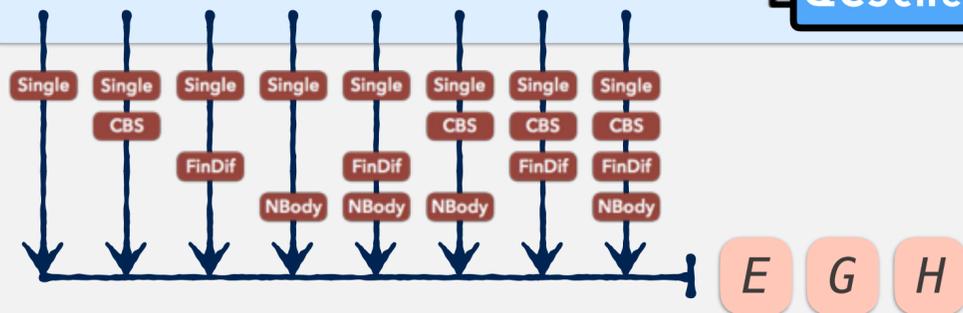
ASM Return analytic energy, gradient, or Hessian.

DISTRIBUTED DRIVER

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



- 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.



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.

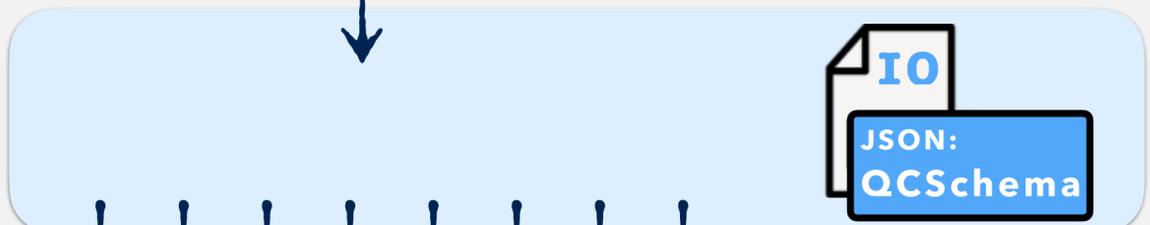
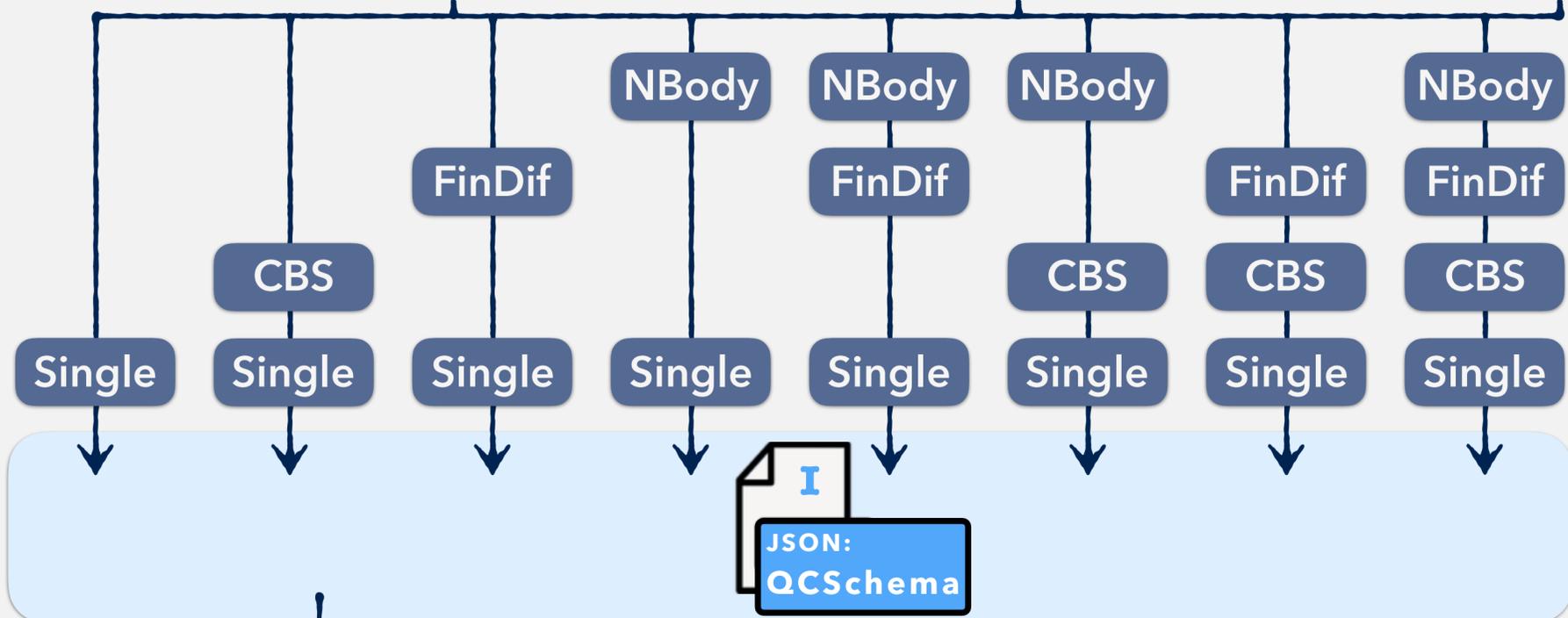
class **SingleResult** ():

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

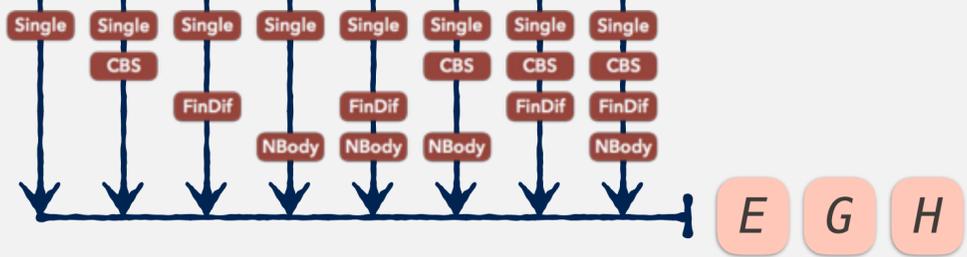
ASM Return analytic energy, gradient, or Hessian.

DISTRIBUTED DRIVER

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



- 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.



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

```
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):

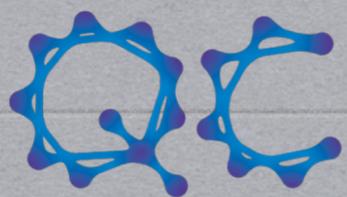
    dimer = psi4.geometry("""0 0 0 0\n H 1 0 0\n H 0 1 0\n --\n 0 3 3 3\n H
dimer.update_geometry()
dimer.set_geometry(psi4.core.Matrix.from_array(coords))

    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=`



(py37) loriab@ariadne:~/Users/loriab/linux/psihub/hrw-labfork/objdir14: (recursive) `ll`

psilocaluser@...: ~/theoryfs2/ds/glick/diatomics: `lr hf`

psilocaluser@...: ~/theoryfs2/ds/glick/diatomics: `lr hf/demo/`

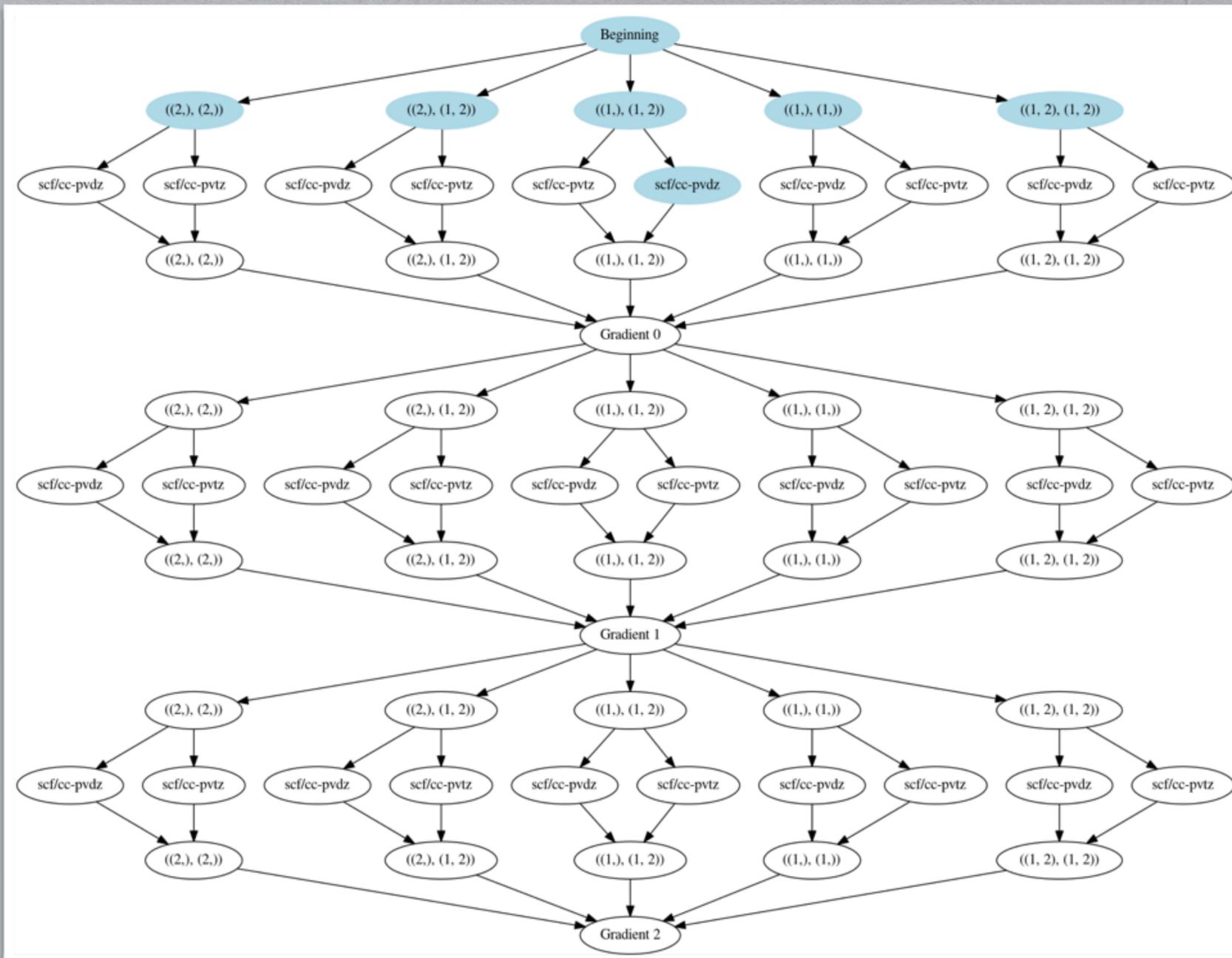
psilocaluser@...: ~/theoryfs2/ds/glick/diatomics: `lr bh/`

psilocaluser@...: ~/theoryfs2/ds/glick/diatomics: `lr bh/demo.py ^C`

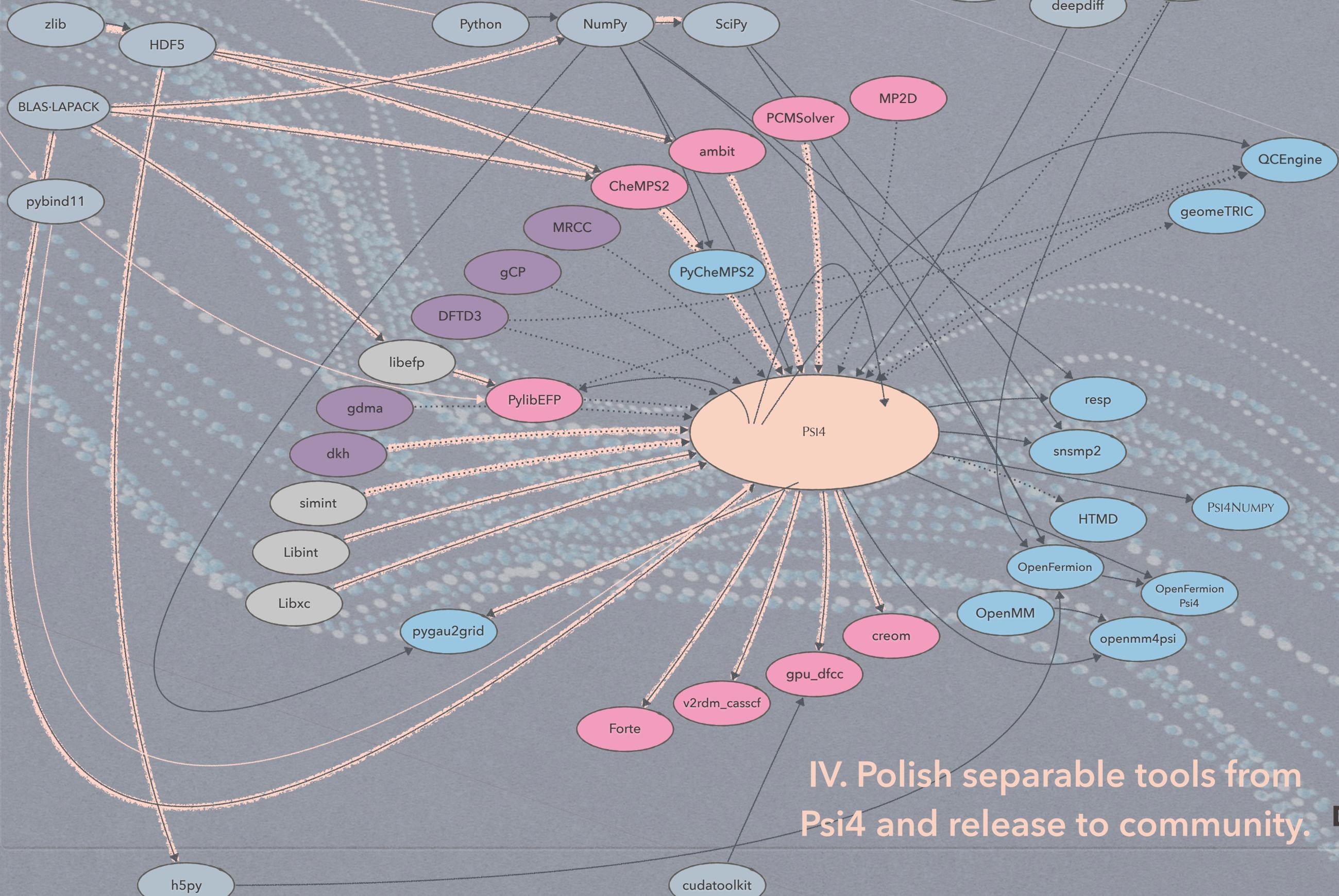
psilocaluser@...: ~/theoryfs2/ds/glick/diatomics: `vi bh/demo.py`

psilocaluser@...: ~/theoryfs2/ds/glick/diatomics: `vi bh/nohup.out`

psilocaluser@...: ~/theoryfs2/ds/glick/diatomics: `ll`



UPSTREAM



DEP is req'd RT dep'd'cy of TGT

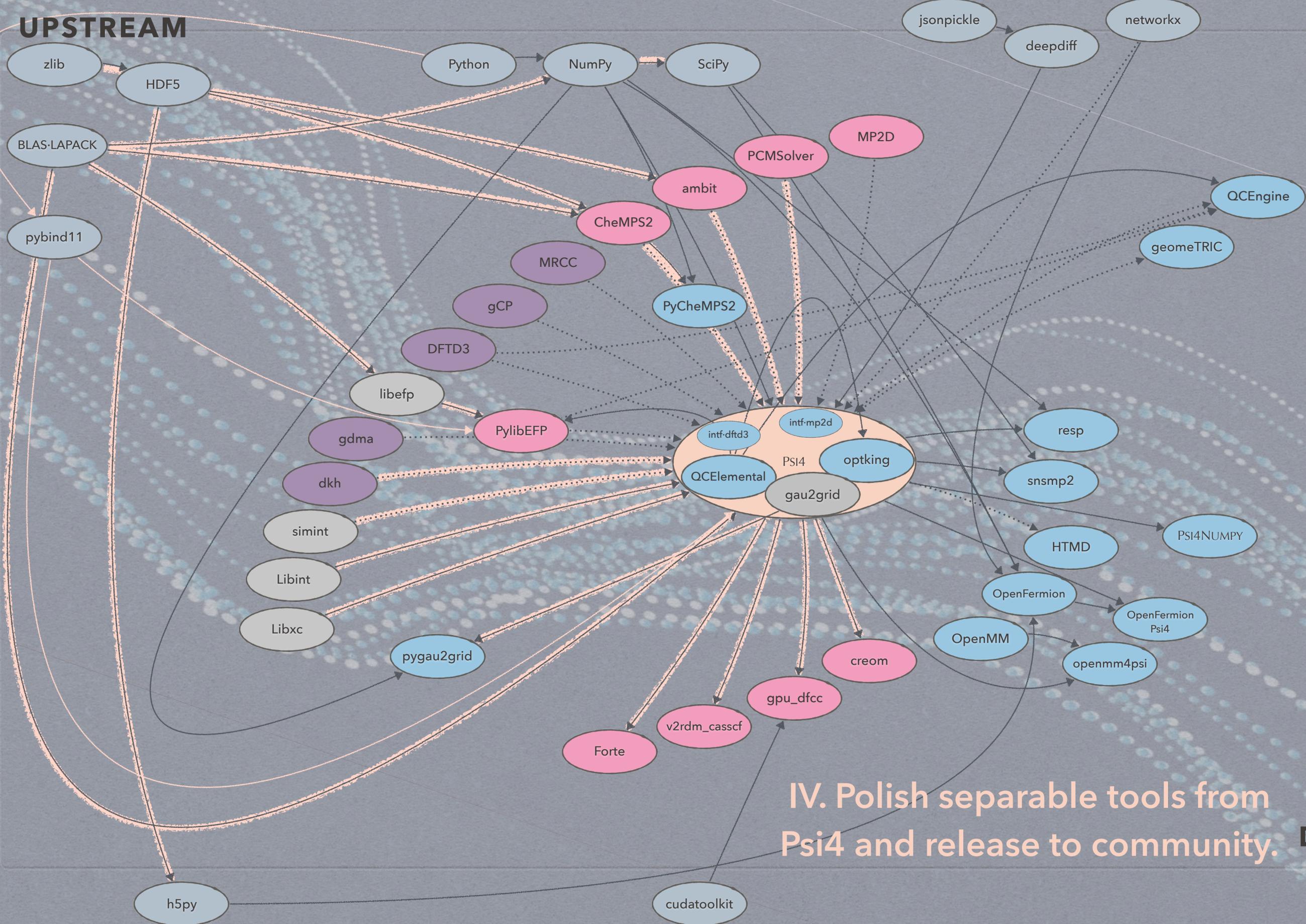
DEP is opt'l RT dep'd'cy of TGT

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



DEP is req'd **RT**
dep'd'cy of **TGT**

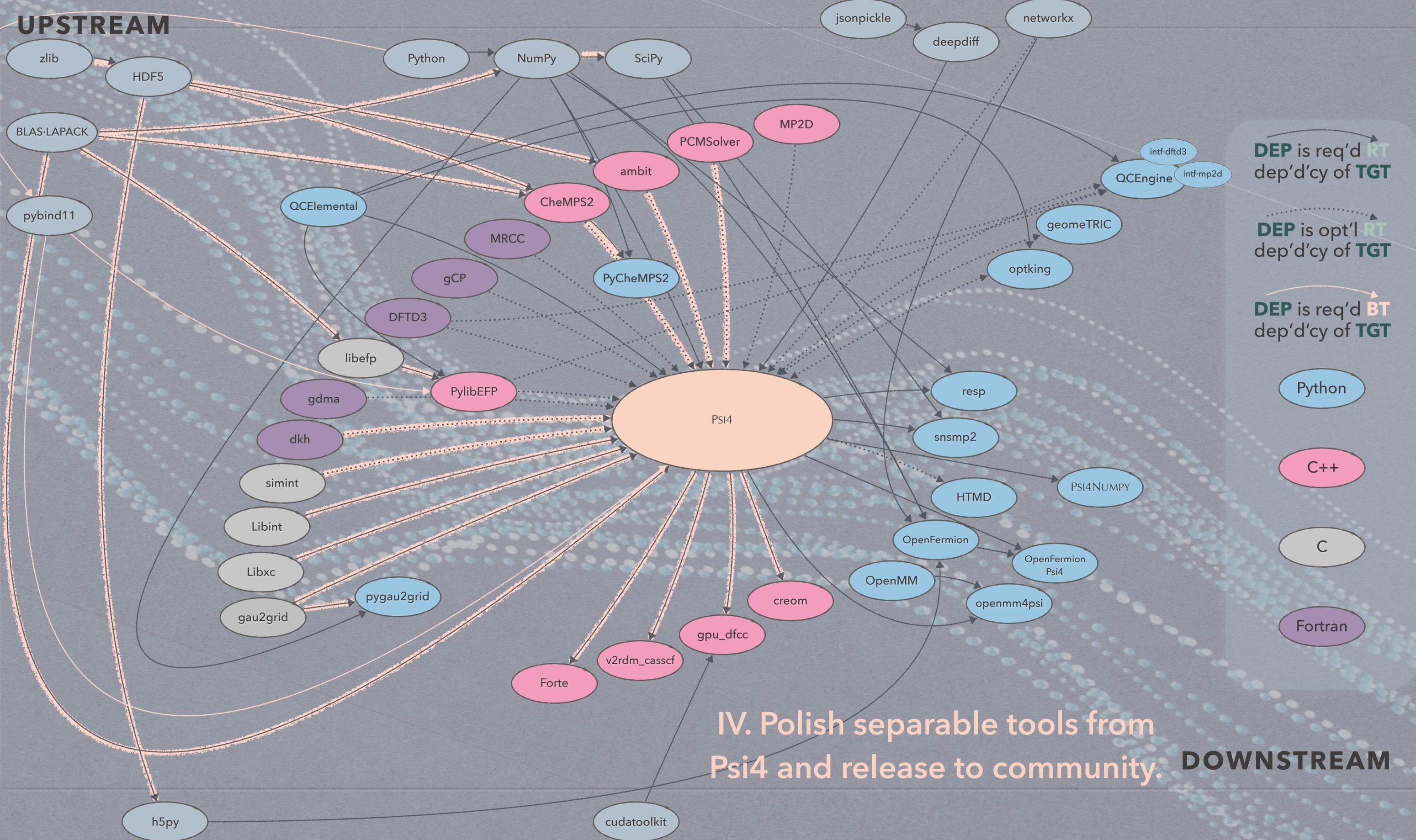
DEP is opt'l **RT**
dep'd'cy of **TGT**

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



DEP is req'd **RT**
dep'd'cy of TGT

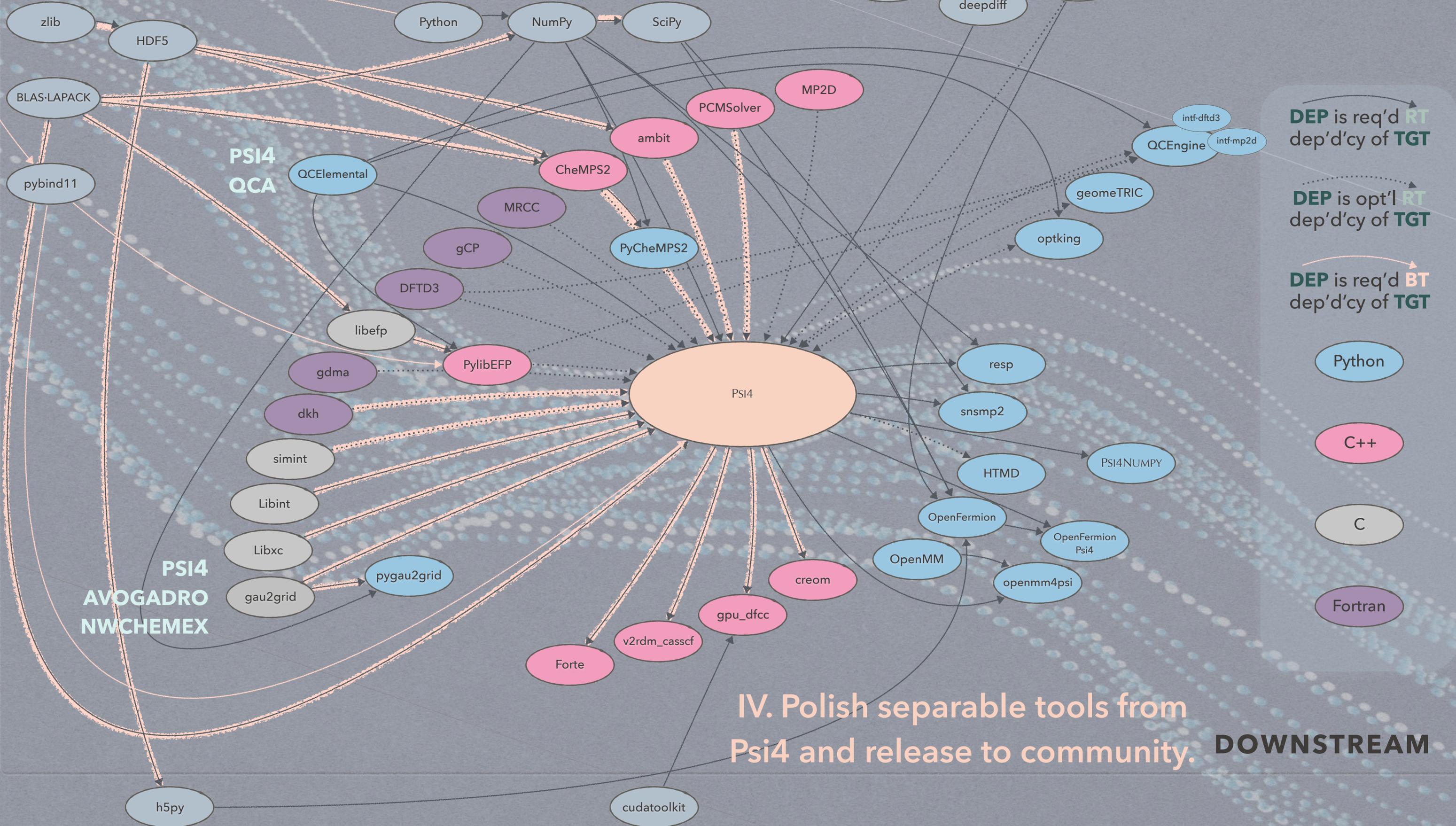
DEP is opt'l **RT**
dep'd'cy of TGT

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



IV. Polish separable tools from Psi4 and release to community. **DOWNSTREAM**

QCDB DEMO

4-PROGRAM FOCAL POINT

(p4dev3/) ps1localuser@bash:ps1net:/home/ps1localuser/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',
})
```

```
# ccstdq 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-ccstdq/cc-pVTZ', return_wfn=True)
E_ccstdtq = float(jrec['qcvars']['CCSDTQ TOTAL ENERGY'].data)
qcdb.set_options({'cfour_dropmo': None})
dE_ccstdtq[i] = E_ccstdtq - E_ccsdpt
print(f'~~~ CCSDTQ Correction={dE_ccstdtq[-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.diatom.anharmonicity(R_arr, E_tot_fci)
```

QCDB DEMO

4-PROGRAM FOCAL POINT

```
(p4dev3/) ps1localuser@bash:ps1net:/home/ps1localuser/gits/qcddb/demo2: (modernps12) time python zach.py
```



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 Simonett
NIH



Ed Hohenstein
CCNY



Roberto Di Remigio
Tromso



Ugur Bozkaya
Hacettepe

PSI4

QCA & QCDB ACKNOWLEDGEMENTS



QCARCHIVE



Daniel Smith
MoSSI



Levi Naden
MoSSI



Theresa Windus
Iowa State

NWCHEM

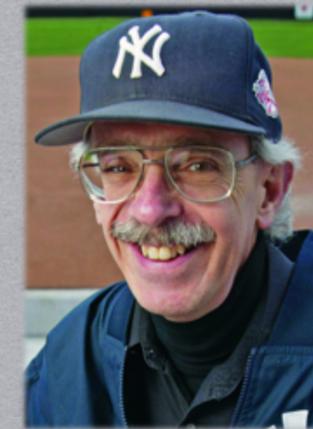


Jiyoung Lee
Iowa State



Annabelle Lolinco
Iowa State

GAMESS



Mark Gordon
Iowa State



Nuwan de Silva

CFOUR



John Stanton
UFL



Devin Matthews
UTexas



David Sherrill
GaTech

PSI4



Daniel Smith
MoSSI



Zach Glick
GaTech

DEMOS



Asim Alenaizan
GaTech



Zach Glick
GaTech

QCA & QCDB ACKNOWLEDGEMENTS



QCARCHIVE



Daniel Smith
MoSSI



Levi Naden
MoSSI



Theresa Windus
Iowa State

NWCHEM

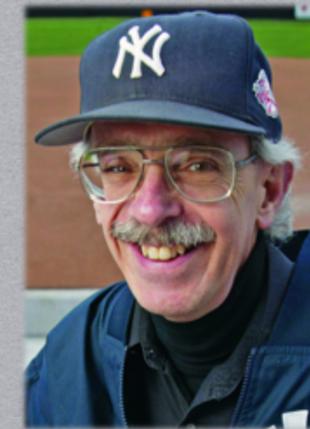


Jiyoung Lee
Iowa State



Annabelle Lolinco
Iowa State

GAMESS



Mark Gordon
Iowa State



Nuwan de Silva

CFOUR



John Stanton
UFL



Devin Matthews
UTexas



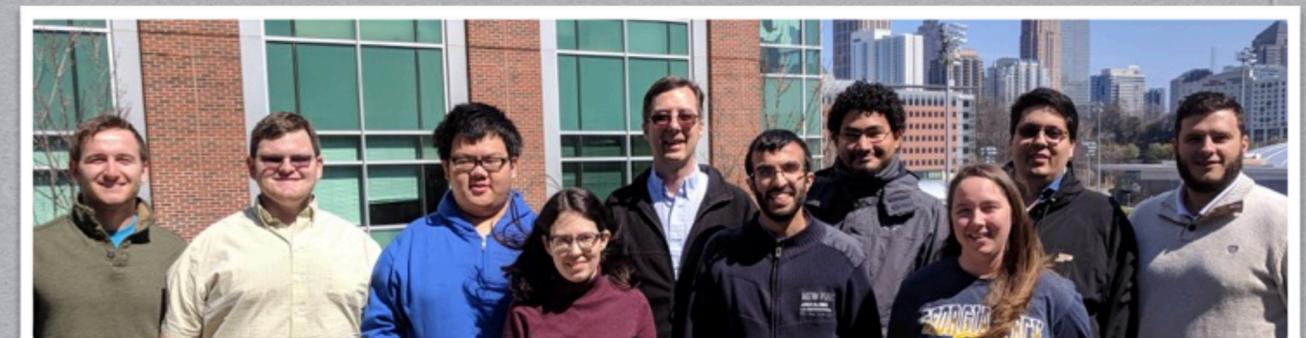
David Sherrill
GaTech

PSI4



Daniel Smith
MoSSI

SHERRILL GROUP



QCA & QCDB ACKNOWLEDGEMENTS



Daniel Smith
MoSSI



Levi Naden
MoSSI



Theresa Windus
Iowa State



Ben Pritchard
MoSSI

QCA & QCDB ACKNOWLEDGEMENTS



Daniel Smith
MolSSI



Levi Naden
MolSSI



Theresa Windus
Iowa State



Ben Pritchard
MolSSI

