

2023 PSI4 ORDNANCE SURVEY



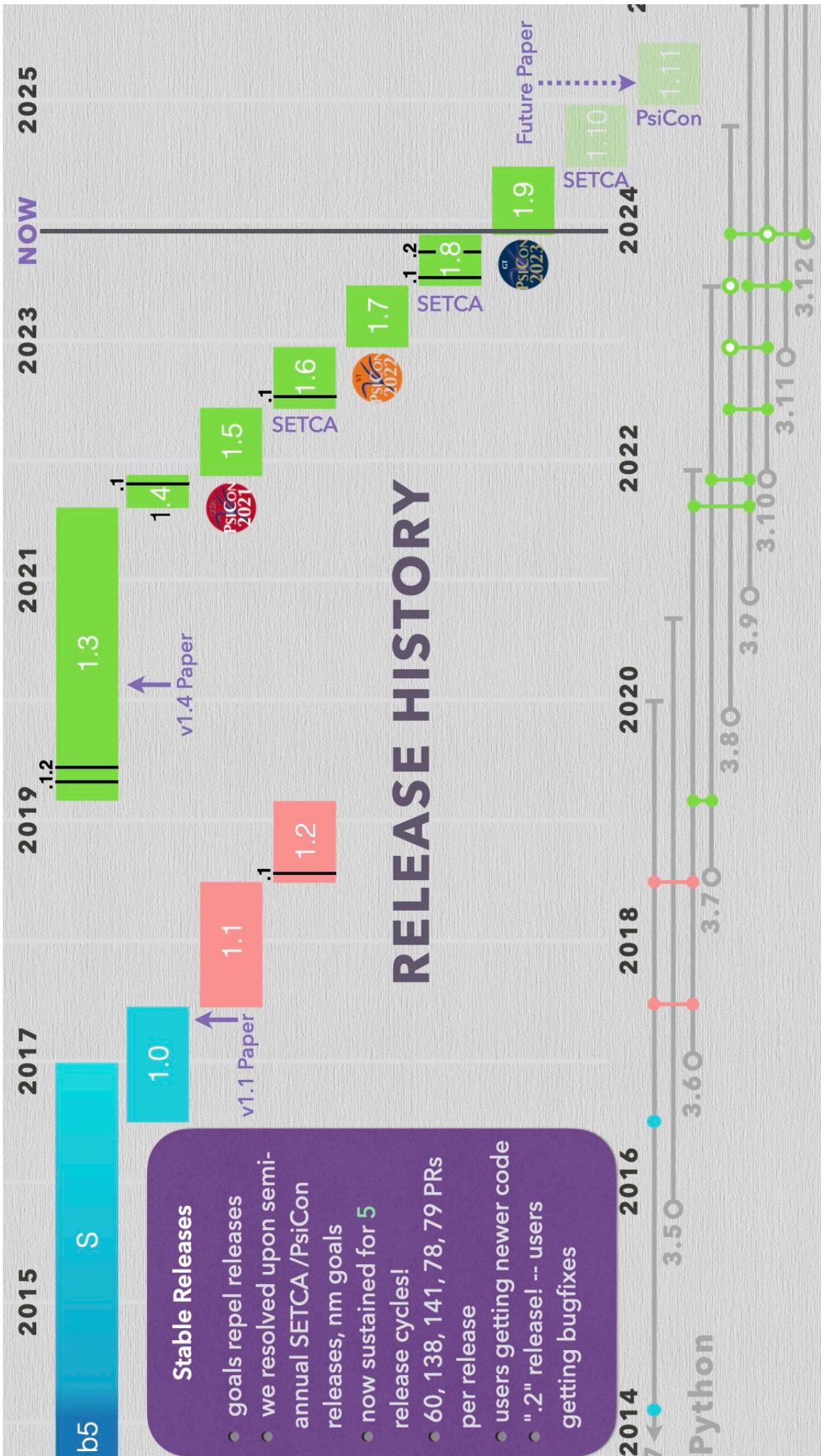
LORI A. BURNS
PSICON 2023, ATLANTA, GA
8 DECEMBER 2023

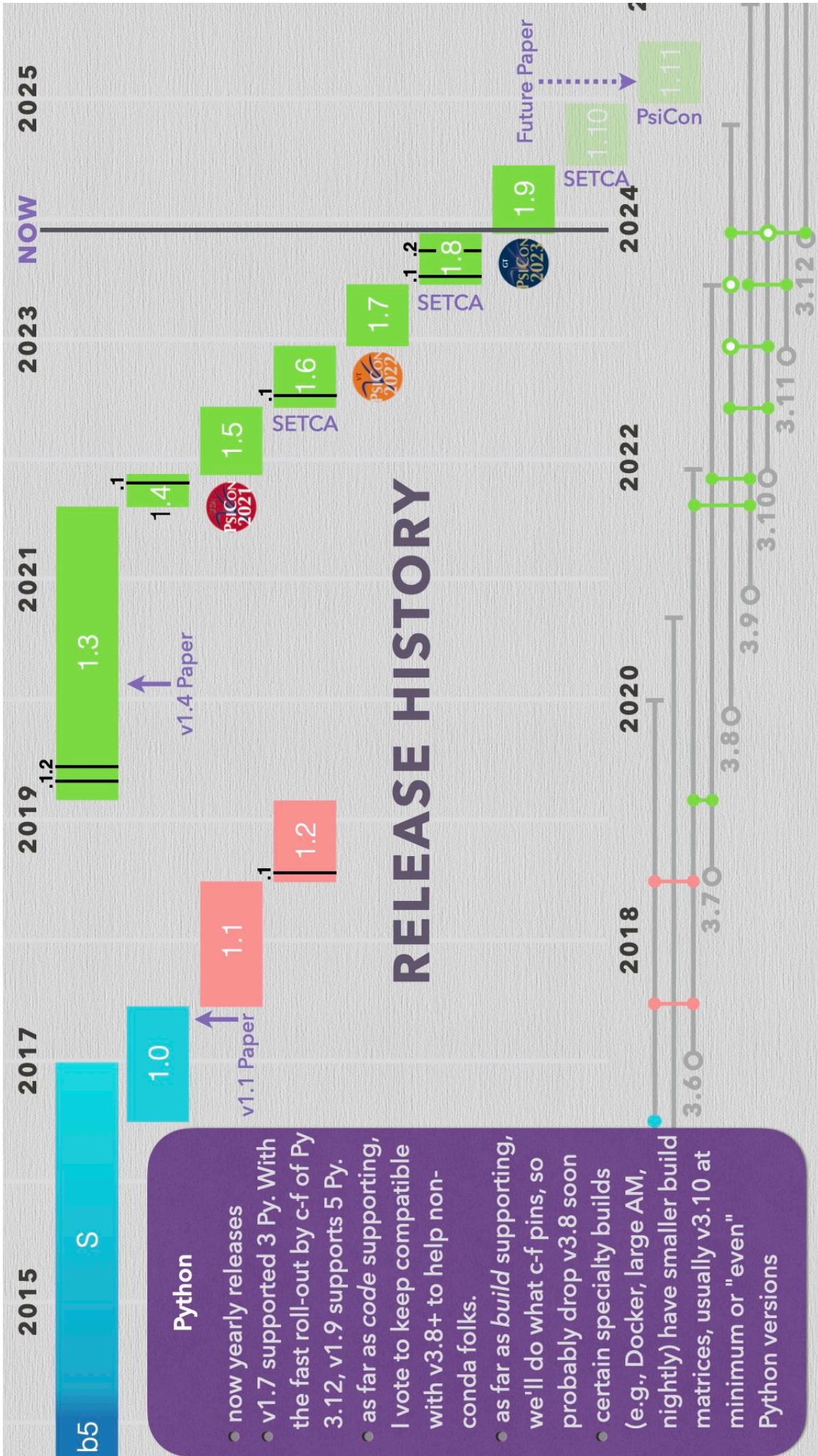


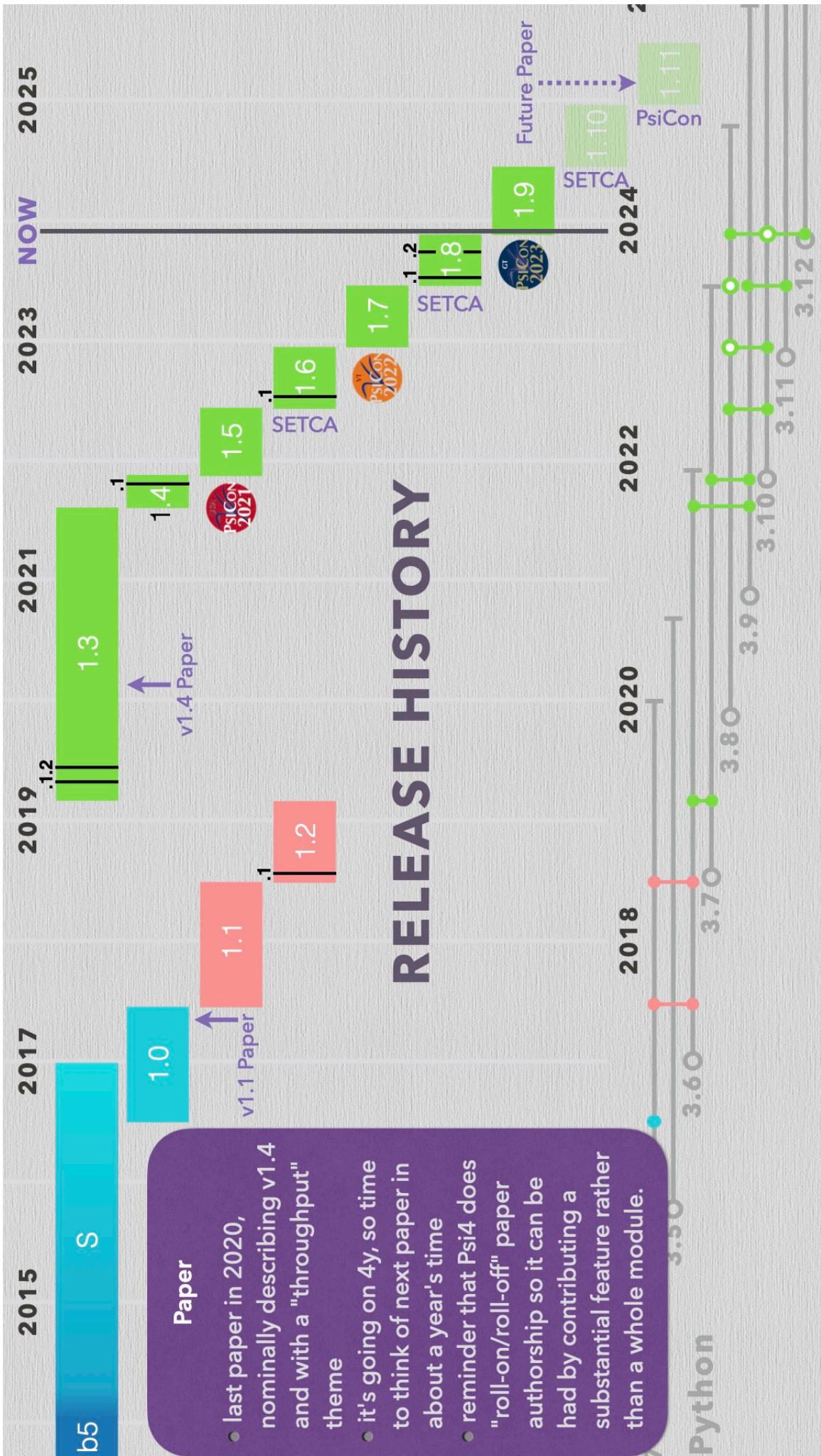
Now Available

1.9.0

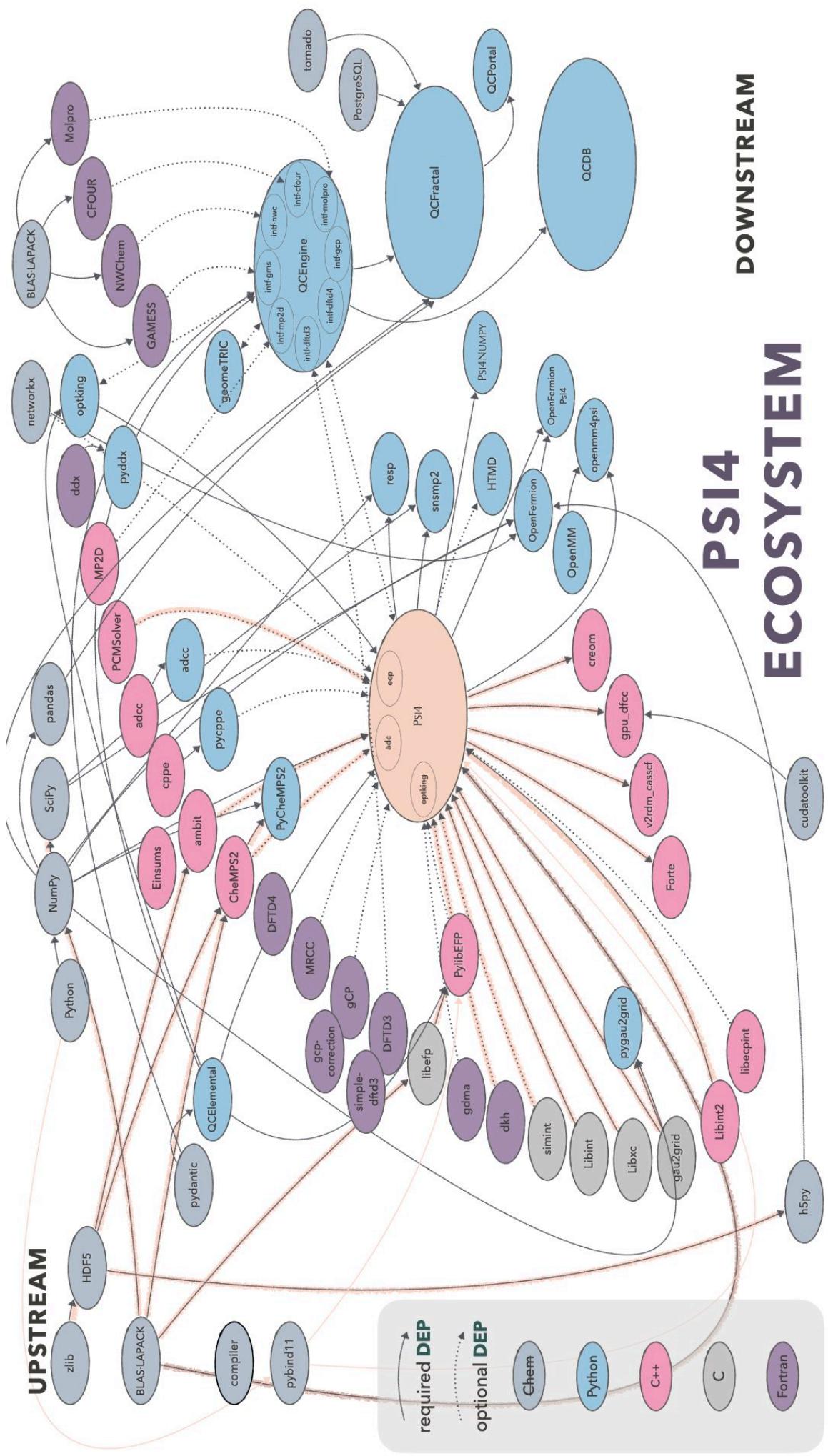
6 December 2023



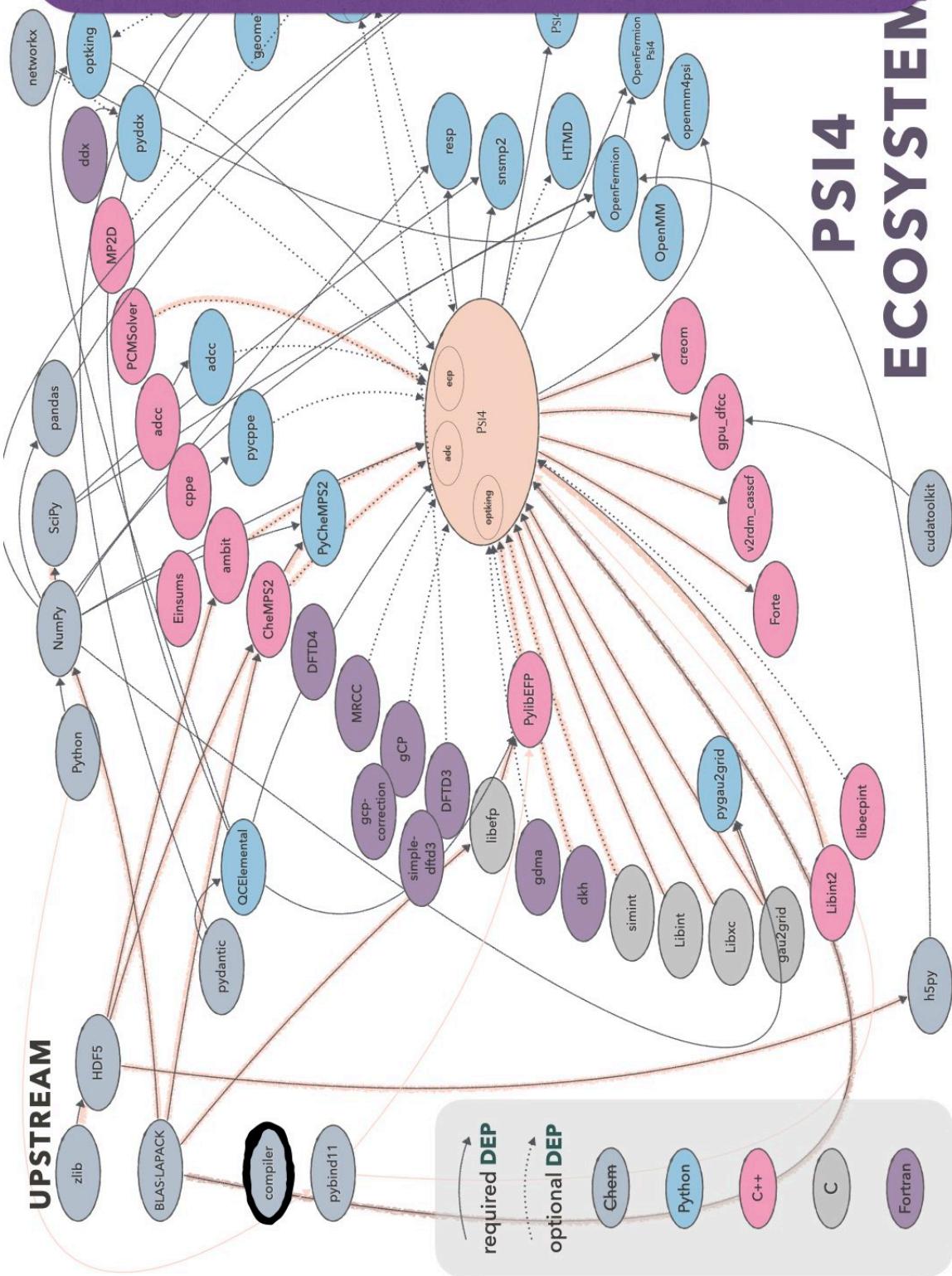


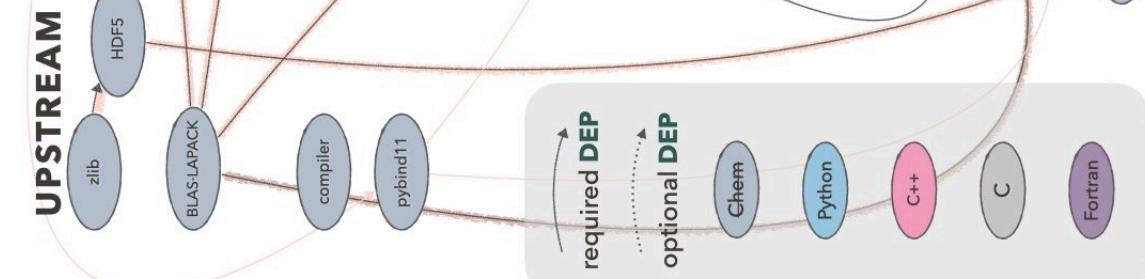


PSI4 ECOSYSTEM

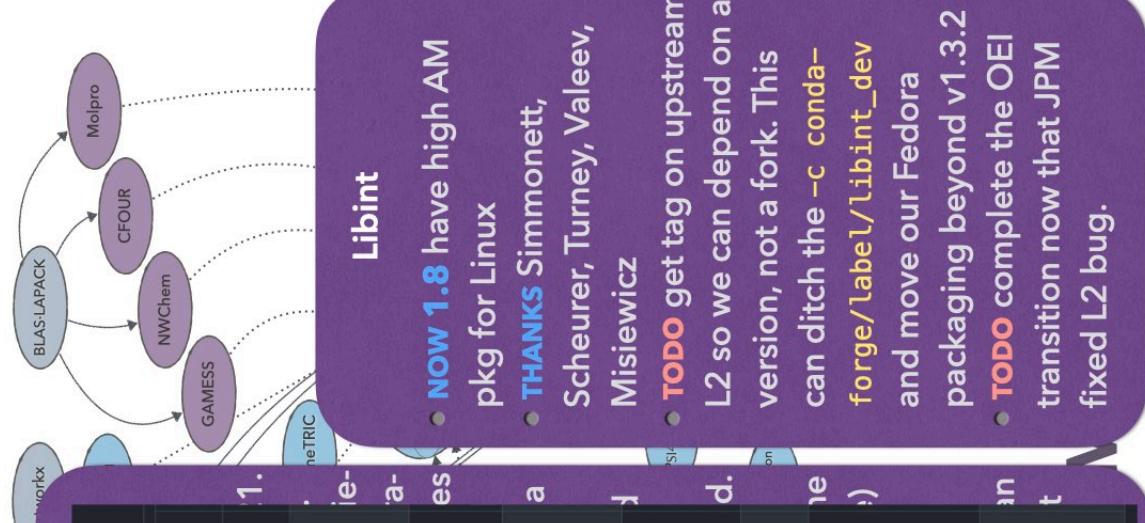


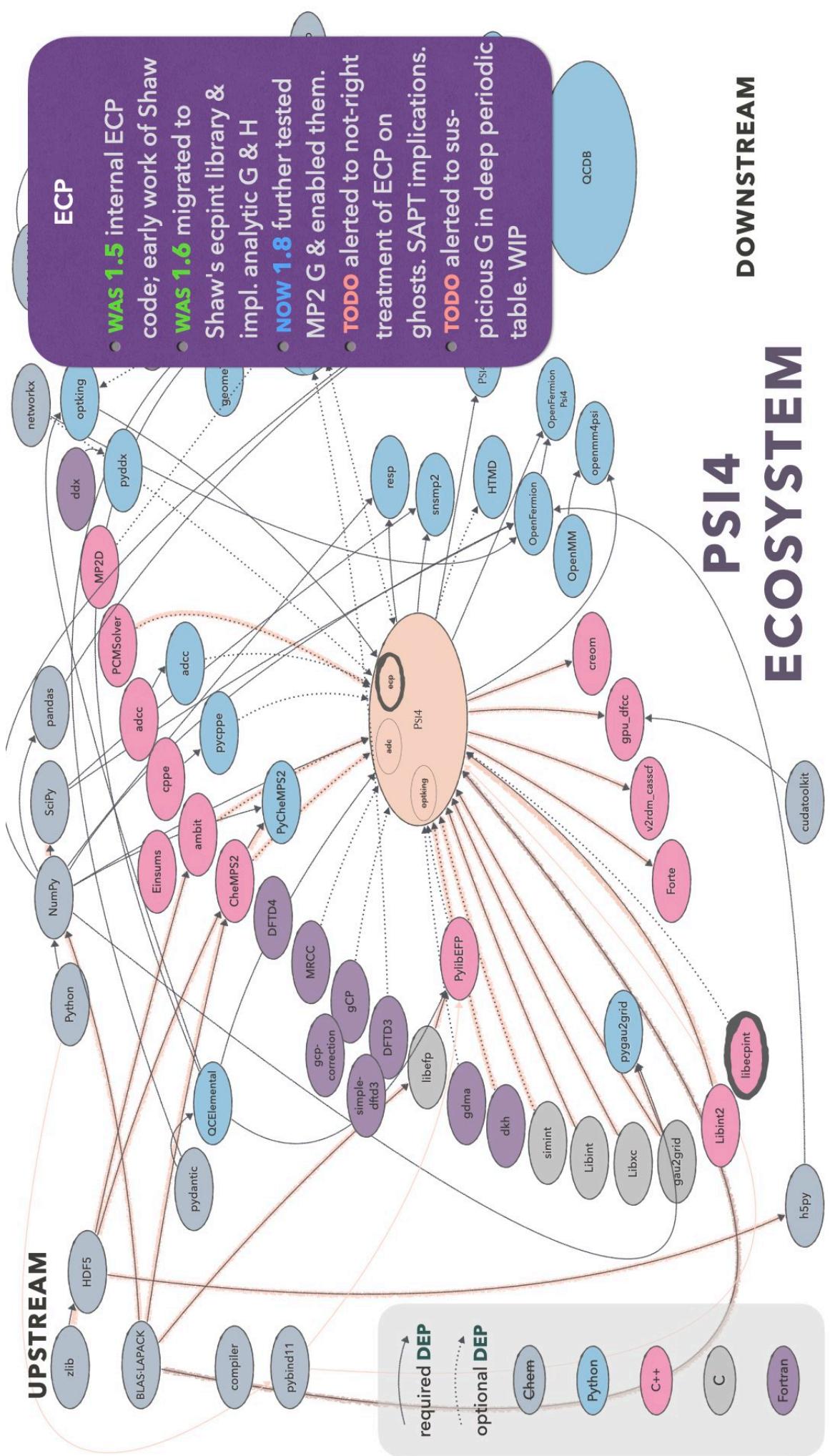
PSI4 ECOSYSTEM

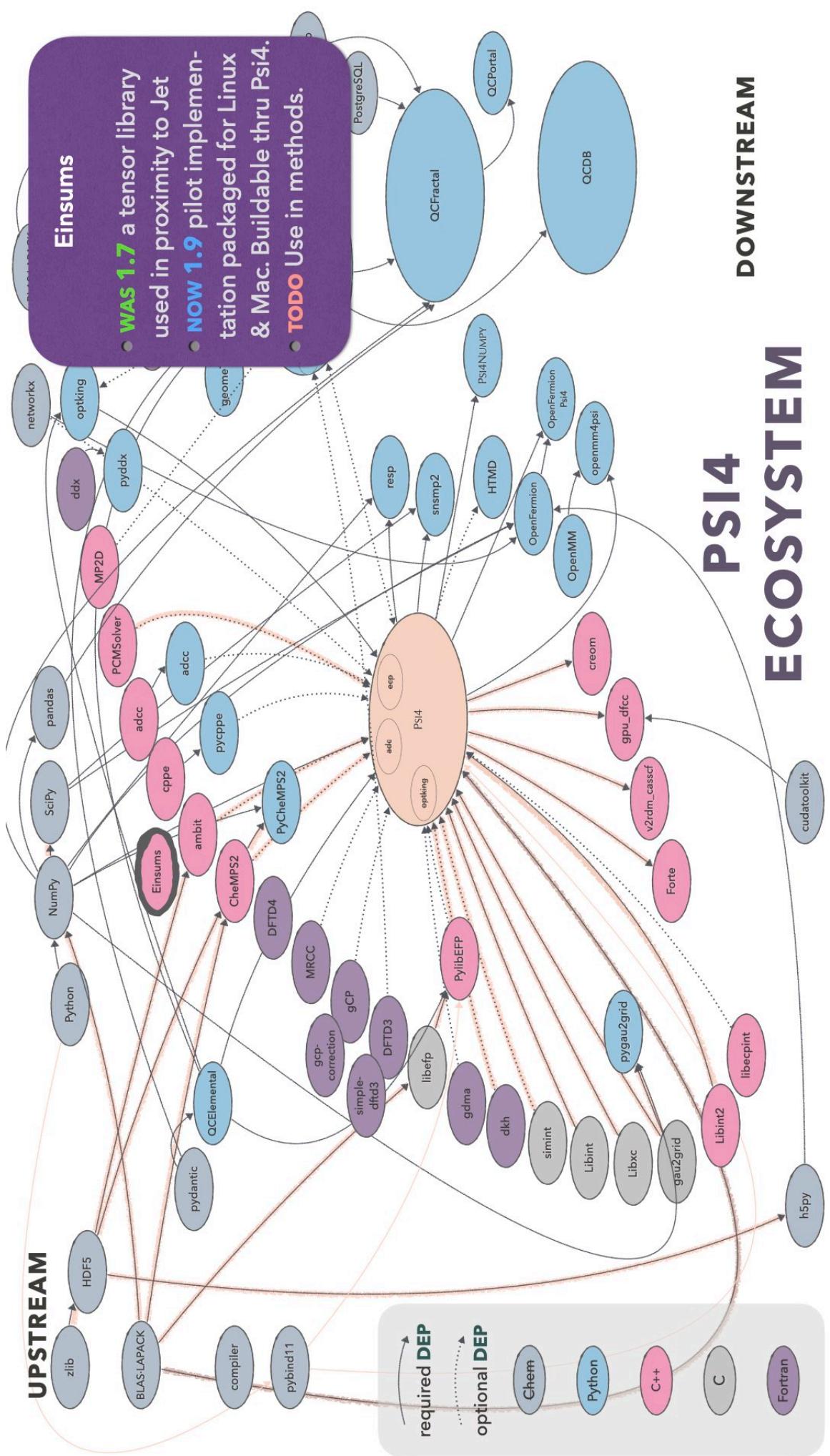




step	status	libint ver & branch	Psi4	tarball[1]	order	component style
1 ^[3]	longstanding L1	L1 evaleev:5c89451	v1.3	—	gss	5
2 ^[4]	TEI L2	loriaib12cmake evaleev/libint#148	20Nov20, after #1721, v1.4, 1.5	L: 7-7-4-7-7-5_1, MW: 5-4-3-6-5-4_1	gss	95
3 ^[5]	OEI L2	ditto step 2	11Mar22, after #2388	L: 5-4-3-6-5-4, mm4f12ob2 , MW: 5-4-3-6-5-4, mm4fob2	gss	95
B ^[6]	upstream L2 cmake	loriaib:new-cmake-harness-lab-b1 evaleev/libint#233	23Mar22, after #2413, v1.6, v1.7	5-4-3-6-5_4_mm4f12ob2.tgz	gss	er1_c4_d1_15
C ^[2]	McMurchie Davidson	any	31Mar22, after #2414, v1.6, v1.7		sss	95
A ^[7]	standardize ordering	ditto step B	#2537	ditto step B	sss	95
A ^[8]	flex solid-harm ordering	loriaib:new-cmake-2023-take2-b evaleev/libint#259	8May23, after #2861, v1.8, v1.9 (see also A'')	libint-2-7-2-post1-5-4-3-6-5-4_mm4f12ob2_1	sss	er1_c4_d1_15
A ^[9]	flex solid-harm ordering, libtool-based	upstream master any point after evaleev/libint@10cad29	3Dec23, after #3047, v1.9 (see also A'')	libint-2.8.0-dev2-5-4-3-6-5-4_mm10f12ob2.tgz	ss	er1_hhhh_d1

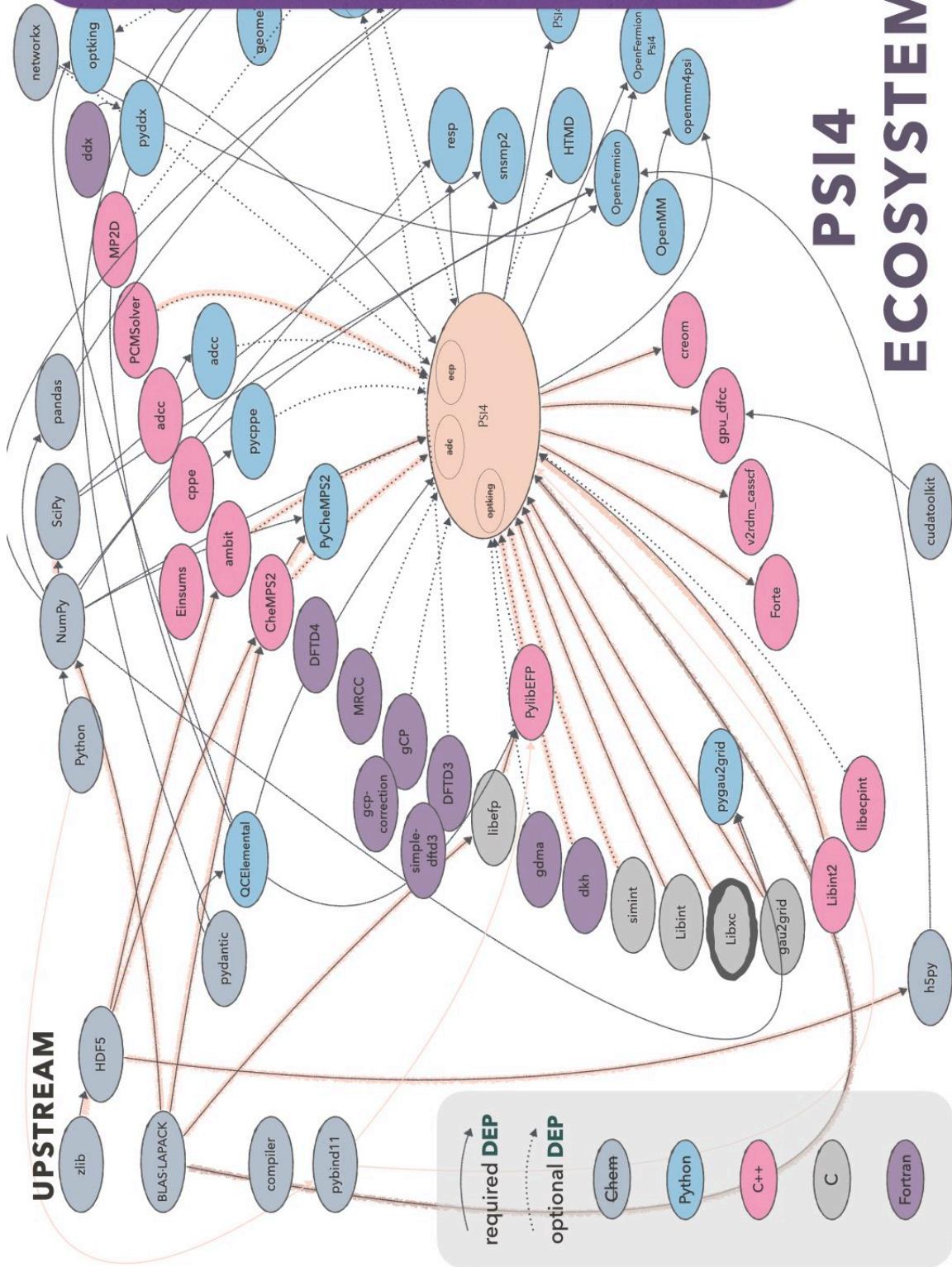




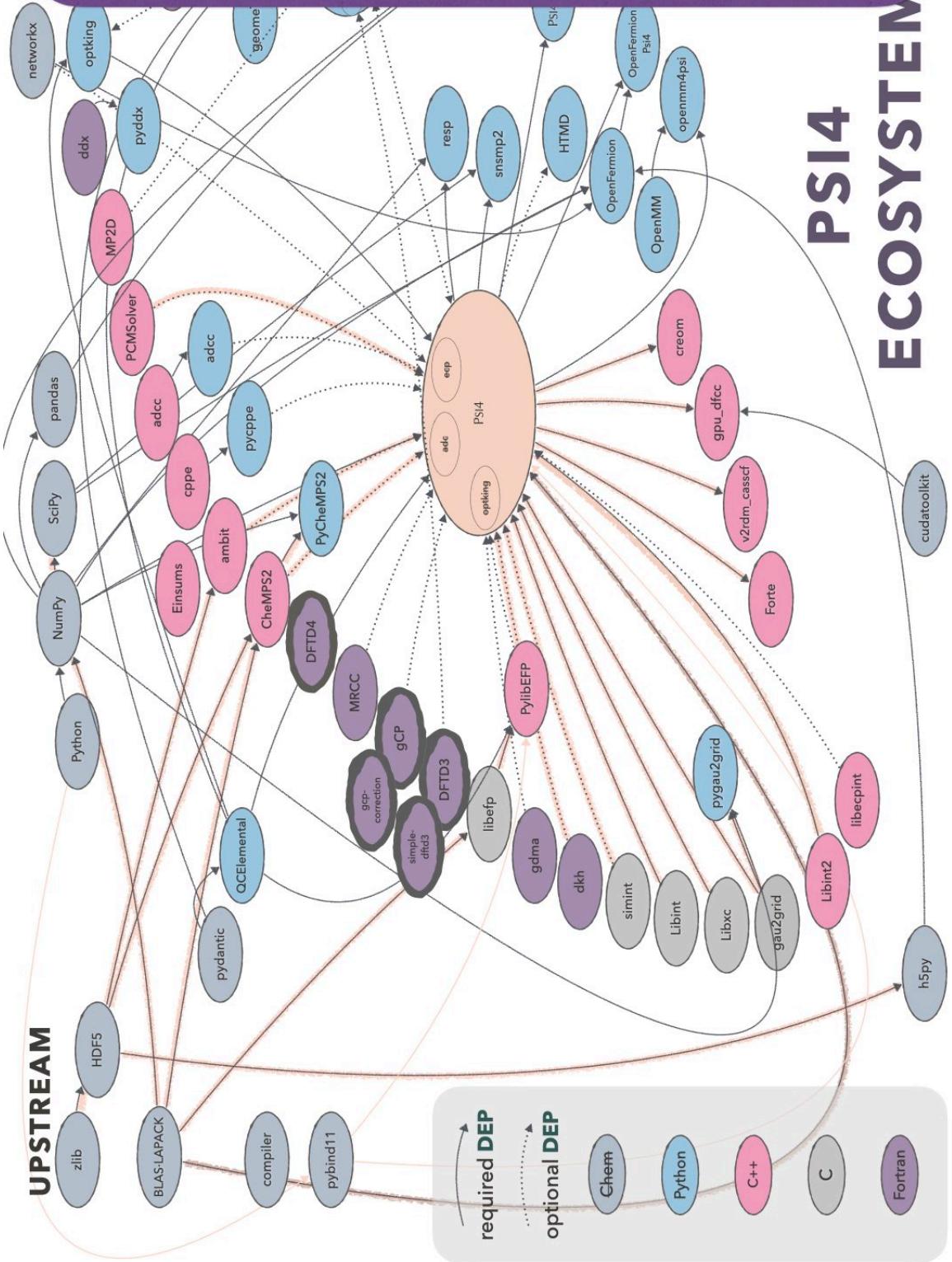


DOWNSTREAM

PSI4 ECOSYSTEM

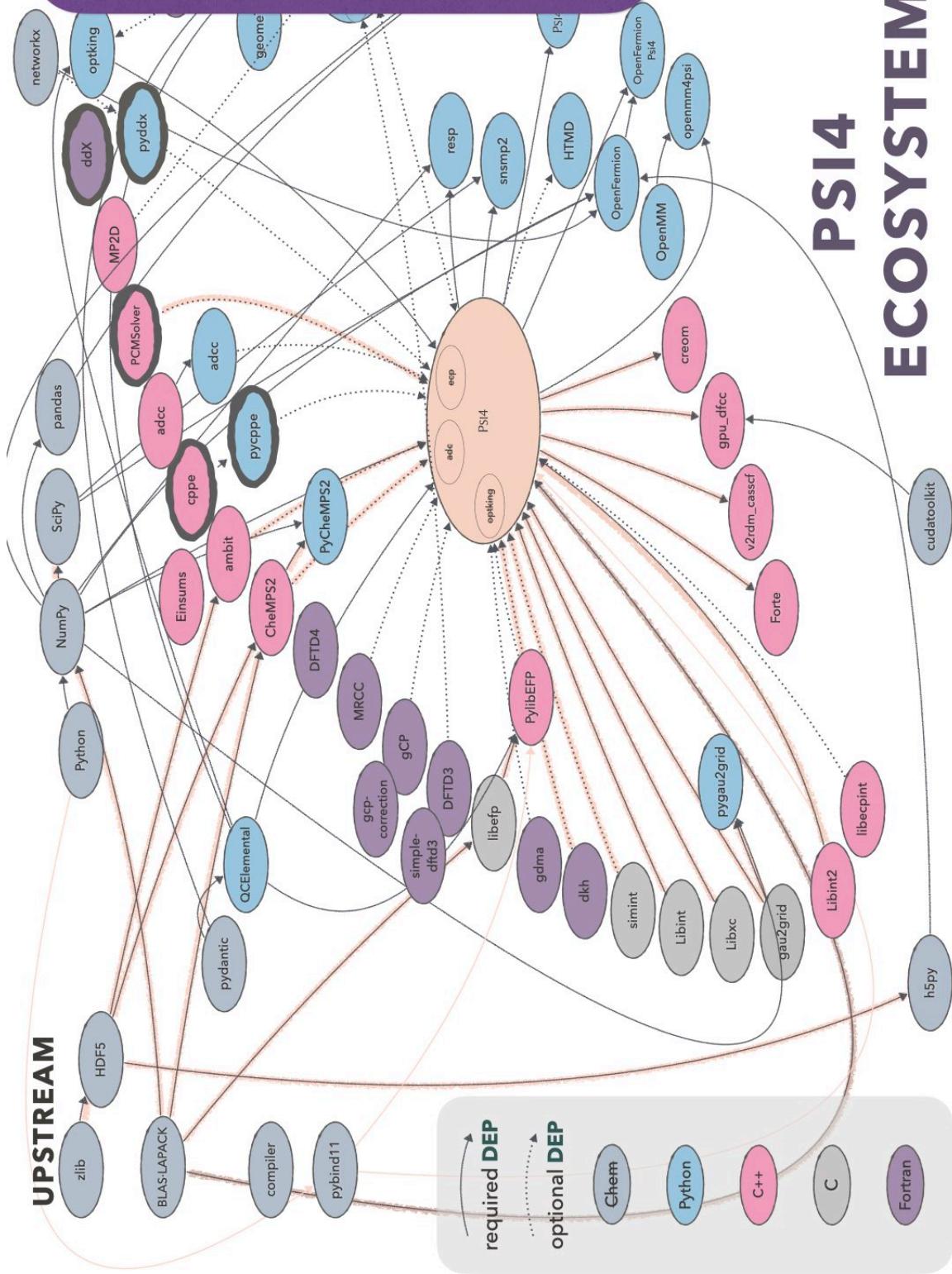


PSI4 ECOSYSTEM



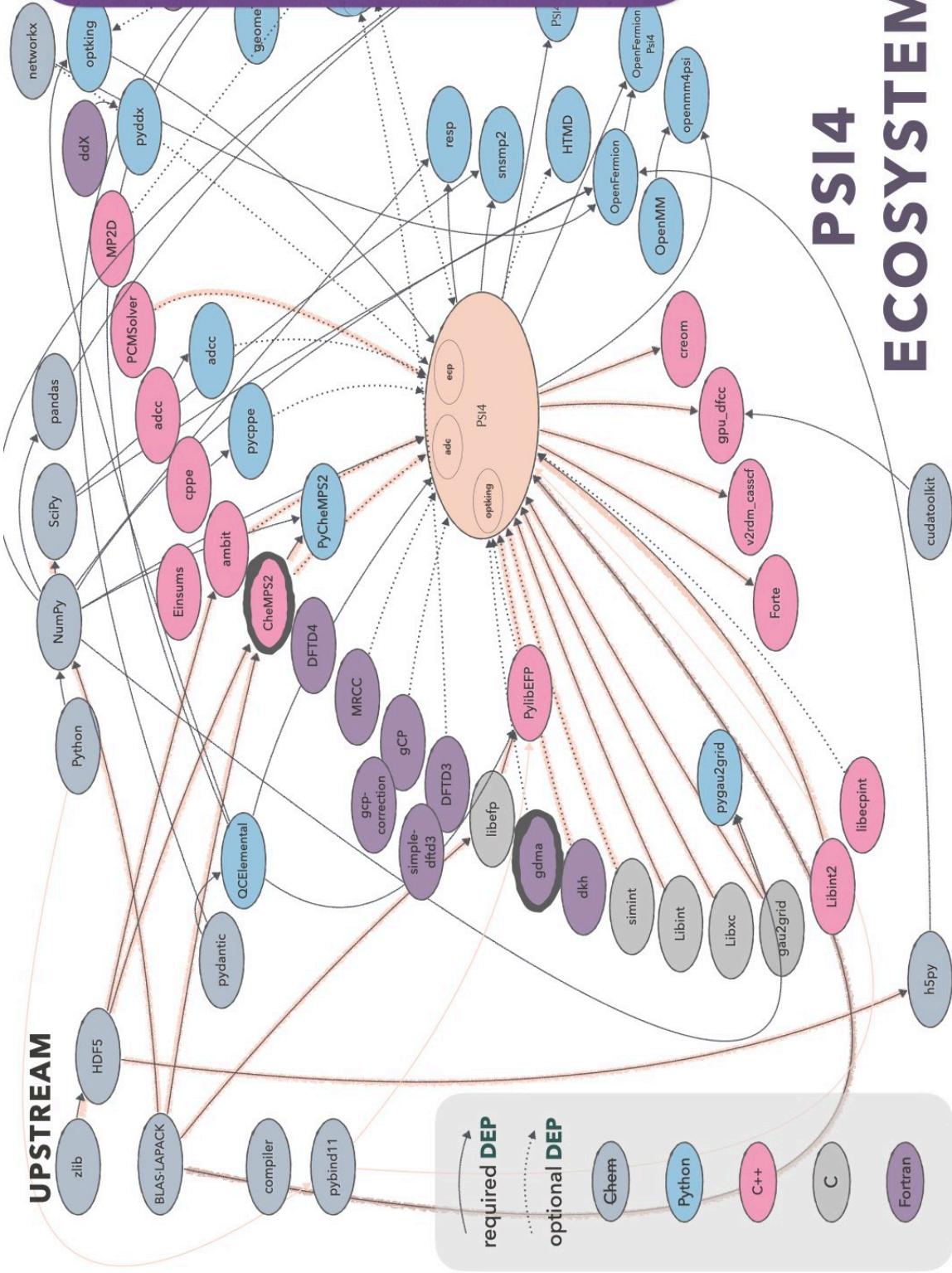
DOWNSTREAM

PSI4 ECOSYSTEM



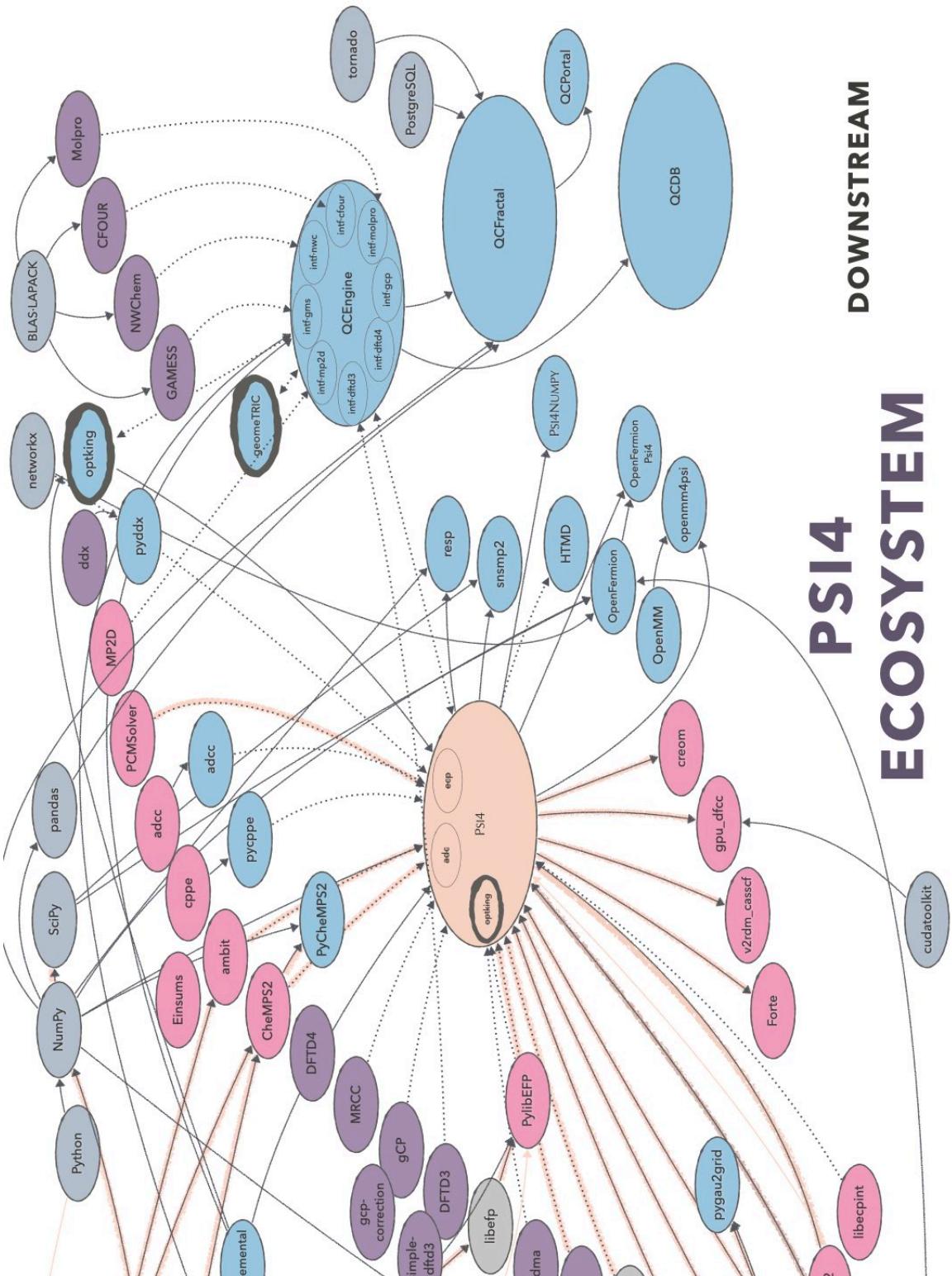
DOWNSTREAM

PSI4 ECOSYSTEM



PSI4 ECOSYSTEM

DOWNSTREAM



Optimizers

- **WAS 1.6** longstanding internal C++ optking optimizer.
- **WAS 1.7** external Python optking as required dependency (Heide, King) **NOW 1.9** remarkable for a big change cause almost no attention beyond a few renamed/lost keyword options. dependency (Heide, King)
- **TODO** new version with accumulated changes.

- **WAS 1.7** Psi4 interface slightly updated to use new v1.0 geomTRIC (L-P Wang).

Fortran

cudatoolkit

h5py

t2

libecpint

Forte

v2rdm.casscf

gpu_dfcc

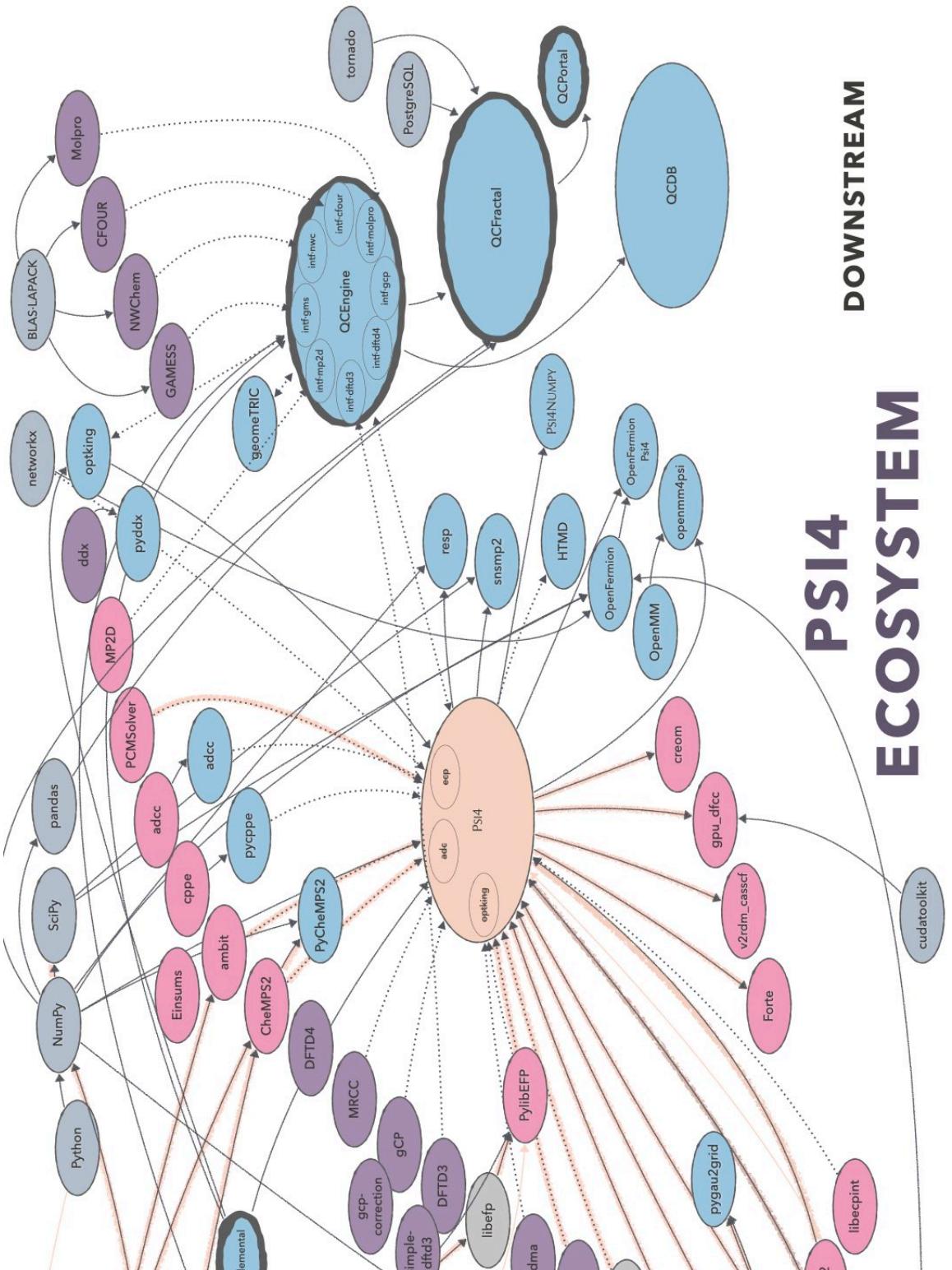
PyGau2Grid

PyTDDFT

PyTDDFT

PSI4 ECOSYSTEM

DOWNSTREAM



QCArchive

- **WAS 1.5** recursive driver
- **WAS 1.6** distributed driver for manybody, composite mtds, & findif useable with legacy QCFractal v0.15.8.
- **WAS 1.7** useable with legacy or "next" rewrite.
- **NOW 1.9** "next" released as v0.52 in Sept. & Psi4 dropped legacy support.
- **NOW 1.9** pydantic v1 & v2-tolerant.
- **TODO** better labeling of jobs & output ctrl from distributed driver.
- **TODO** extract some distributed driver code to enable reuse by QCA.

Fortran

h5py

cudatoolkit

t2

libecpint

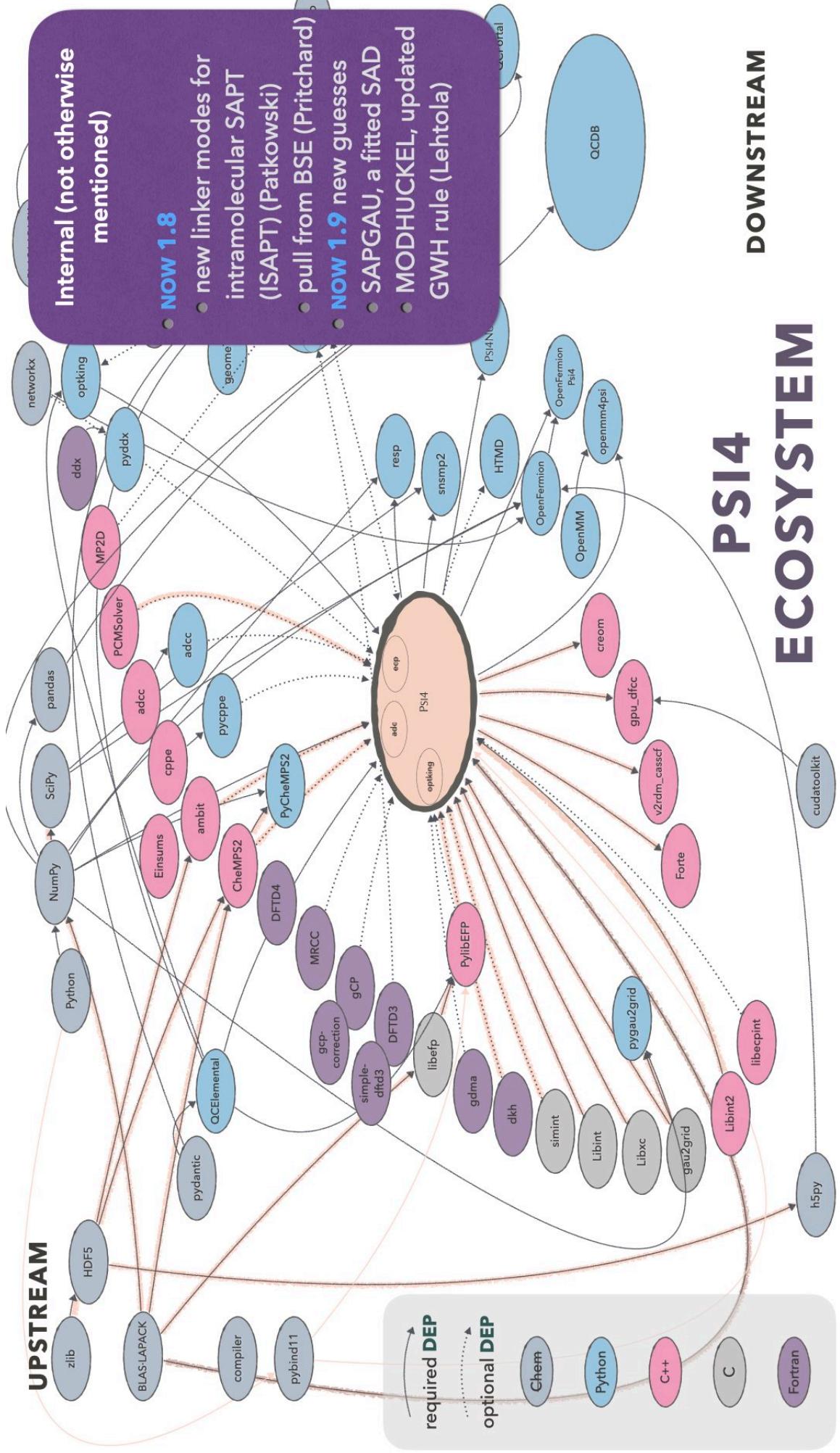
PylibEFP

Pydantic

PyGauZGrid

DOWNSTREAM

PSI4 ECOSYSTEM



CONDA-FORGE TRANSITION

- ## • **PSI4 & CONDA-FORGE CHANNELS**

- **-C PSI4** is longstanding home & a nice one-stop shop for Psi4 end users
 - has a lot of QC projects and is defaults-channel-based
 - **-CONDA-FORGE** is community conda packaging
 - starting with **V1.8**, I moved psi4 there, after adding its dependencies
 - in future, use psi4 channel only for **LARGE AM L2 & Psi4** and for **NIGHTLY** builds. Even then, add `-c conda-forge`

- moved **FOR ME**
 - Psi4 channel has a lot of packages (yellow+green; 83; many unused) needing updates and rebuilding for new Pv, HDE5, clang etc

- QC: a good many (green) already on c-f and maintained by their developers: **SIMPLIDFTD3**, **GCP-CORRECTION**, **DFTD4-PYTHON**, **CPPE**, **GEOMETRIC**, **QCENGINE**, **QCFRACTAL**, **ADCC**, **DDX**, **OPTKING**, **GEOMETRIC**, **OPENFERMION**, **GAU2GRID**, **LIBXC**

- non-QoS: defaults channel supports **3K** packages. c-f supports **23K** packages

- **CO-MAINTAINERS** on packages possible

- **AUTOTICK** and migrations ease maintenance

- moved FOR YOU

- within psi4, solving a full **ECOSYSTEM** is involved.

- but for projects **DOWNSTREAM**, getting a robust environment from multiple channels is harder still and may hit real conflicts, either of version or runtime compatibility. Much easier if we're all on the same channel with common pinnings.

- more **RAPID** compiler, Python, etc. releases

- PATCH releases of Psi4 easy

- moved FOR OS/ARCHITECTURES

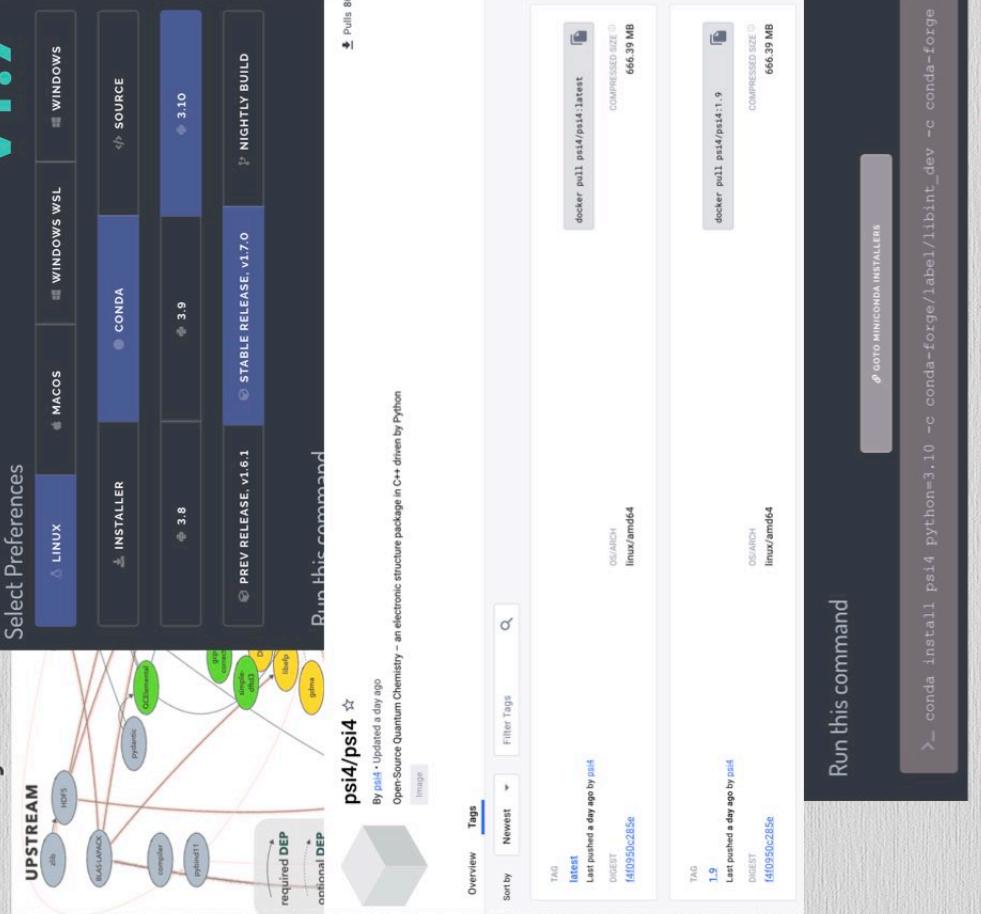
- Builds run in CI, so Linux, Mac (Intel), & Windows all available.

- cross-compilation set up, so Mac (**silicon**) accessible.

- CUDA** also available for experimentation.

- In general:** `conda install psi4 -c conda-forge/libint_dev -c conda-forge`

why did I do this to you?



Get Started with PS14

V.7

RETURN OF THE BUILD HELPER

- packaging is easy compared to tools to help you build Psi4 yourself.
- The trouble is everyone has different, very reasonable, needs
 - no addons vs. I need to know the package name for dftd3
 - want to use Accelerate vs. want MKL
 - want high AM integrals
- just want to know how to tell it where Eigen is
- Solution is two files:
 - YAML of all the dependency information: **codedeps.yaml**
 - script that turns it into well-established intermediates that you can edit: conda env spec and CMake cache file: **conda/psi4-path-advisor.py**
- Run **conda/psi4-path-advisor.py -h** (may take a few seconds)
- Run **conda/psi4-path-advisor.py env** to generate env file. edit and create env.
- Build, taking your chances, -or-
- Run **conda/psi4-path-advisor.py cmake** to generate cache file. edit and configure cmake.
- Run **conda/psi4-path-advisor.py bulletin** for updates like advice on switching to mamba through conda cmd

```
psi4 / devtools / conda-envs / pip /Users/loriab/linux/minicond

# <<< python
set( Python_EXECUTABLE

# <<< cmake
set(CMAKE_PREFIX_PATH
    # set(CMAKE_BUILD_TYPE
    # set(CMAKE_VERBOSE_MAKEFILE

# <<< ninja
# The CMake generator can be set in a cache file but then can't
# set(CMAKE_GENERATOR
# set(CMAKE_GENERATOR

# <<< blas-devel
set(LAPACK_LIBRARIES
)

503      # name: pylibefp
504      # constraint: ">=0.6.2"
505      # target: pylibefp::core
506      # components: shallow
507      # ENABLE_libefp
508      # cond:
509      # channel: conda-forge
510      # name: pylibefp
511      # note: "0.6.2 provides windows arch"
512      # constraint: null
513      # cmake:
514      # ENABLE_libefp: true
515      # libefp_DIR:
516      # unix: ${CONDA_PREFIX}/share/cmake/libefp
517      # win-64: ${CONDA_PREFIX}/library/share/cmake/libefp
518      # //CMAKE_FIND_PACKAGE_pylibefp: true
519      # //CMAKE_DISABLE_FIND_PACKAGE_pylibefp: true
520      # cmake.note: "Primarily OFF runtime detected. With package present, CMake
521      # enabling
522      # - project: mrcc
523      # use:
524      # added: "1.0"
525      # required: false
526      # buildable: false
```

```

conda/psi4-path-advisor.py -h
usage: psi4-path-advisor [-h] [-v] {env,conda,cache,cmake} ...

Dependency, Build, and Run path advisor for Psi4.
Mediates file https://github.com/psi4/psi4/blob/master/codedeps.yaml
Run env subcommand. Conda env create and activate. Run cmake subcommand. Build.

=====
(A) black-box usage (copy/paste-able)
=====

# (1) get code from GitHub
git clone https://github.com/psi4/psi4.git && cd psi4
# (2) generate env spec file from codedeps.yaml. "eval $(...)" creates and activates conda env.
eval $(conda/psi4-path-advisor.py env)
# (3) generate cmake cache file from conda env. "eval $(...)" configures and builds with cmake.
eval $(conda/psi4-path-advisor.py cmake)

shows up in p4dev

=====
(B) flexible usage
=====

# (1) get code from GitHub
git clone https://github.com/psi4/psi4.git && cd psi4

# (2.0) consider dependency options
conda/psi4-path-advisor.py env -h
# (2.1) generate env spec file from codedeps.yaml.
conda/psi4-path-advisor.py env -n p4dev310 --python 3.10 --disable addons --lapack openblas
#> conda env create -n p4dev310 -f /home/psi4/env_p4dev310.yaml --solver libmamba && conda activate p4dev310
# (2.2) edit env_p4dev310.yaml to customize software packages.
# (2.3) issue suggested or customized command to create and activate conda env.
conda env create -n p4dev310 -f /home/psi4/env_p4dev310.yaml --solver libmamba && conda activate p4dev310

# (3.0) consider compile options
conda/psi4-path-advisor.py cmake -h
# (3.1) generate cmake cache file from conda env.
conda/psi4-path-advisor.py cmake
#> cmake -S. -GNinja -C/home/psi4/cache_p4dev310.cmake -Bobjdir_p4dev310 && cmake --build objdir_p4dev310
# (3.2) edit cache_p4dev310.cmake to customize build configuration.
# (3.3) issue suggested or customized command to configure and build with cmake.
cmake -S. -GNinja -C/home/psi4/cache_p4dev310.cmake -Bobjdir_p4dev310 -DCMAKE_INSTALL_PREFIX=/path/to/install-psi4 && cmake --build objdir_p4dev310

```

