

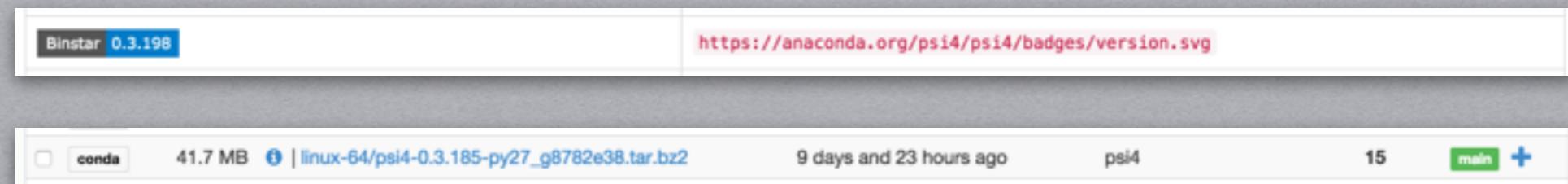
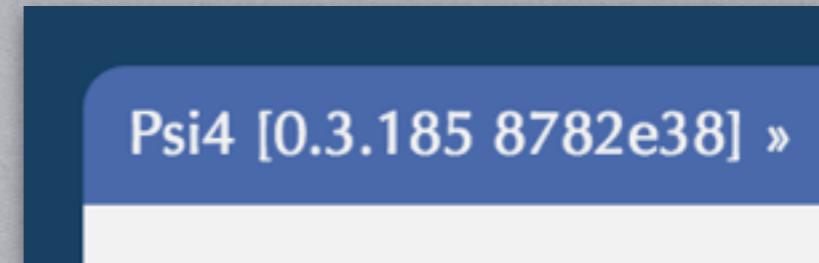
Psi4 UPDATE AND PLANS

LORI A. BURNS

DEVELOPERS MEETING, EMORY UNIVERSITY
13 NOVEMBER 2015

VERSIONING

- project name: Psi4
- ~current version: 0.3.185 8782e38
- major.minor.patch incrementation
 - major: 0 for beta, 1 for release, 2 for API change/next paper?
 - minor: incremented by tags upon feature readiness
 - patch: incremented by git commit
- HAVE: automatic one-to-one mapping btwn git hash and sortable version
- HAVE: binary, docs, and output header following this scheme



Psi4 1.0.0 TARGET

- Let us finish or correct anything that may be deceptive to the user
 - basis set names, high-zeta aux defaults
 - fix Ugur's tests, suppress other failing ones
 - note failing samples in input
- Let us finish anything that just needs tidying or a last step
 - docs refresh from this summer
 - finish CheMPS2 docs
 - decide on PCMSolver
 - reseparate int/deriv AM for compilation
 - review AJ's changes
 - reimplement SCS-DH?
 - review methods list for mixing mtd and algorithm- bad
- Then let us declare 1.0.0
- SAPT
 - energy terms into psivars
 - alpha scaling, MP2 correction, overall procedures
 - reselect defaults among Ed/Rob/Jerome codes
- basis sets
 - aux basis sets lack names, hence mysterious in output file
 - 6Z orbital falls back on QZ fitting. needs warning or halt
 - needs user-defined and programmer's use basis docs
- finish CheMPS2 docs
- approve what AJ did with SCF

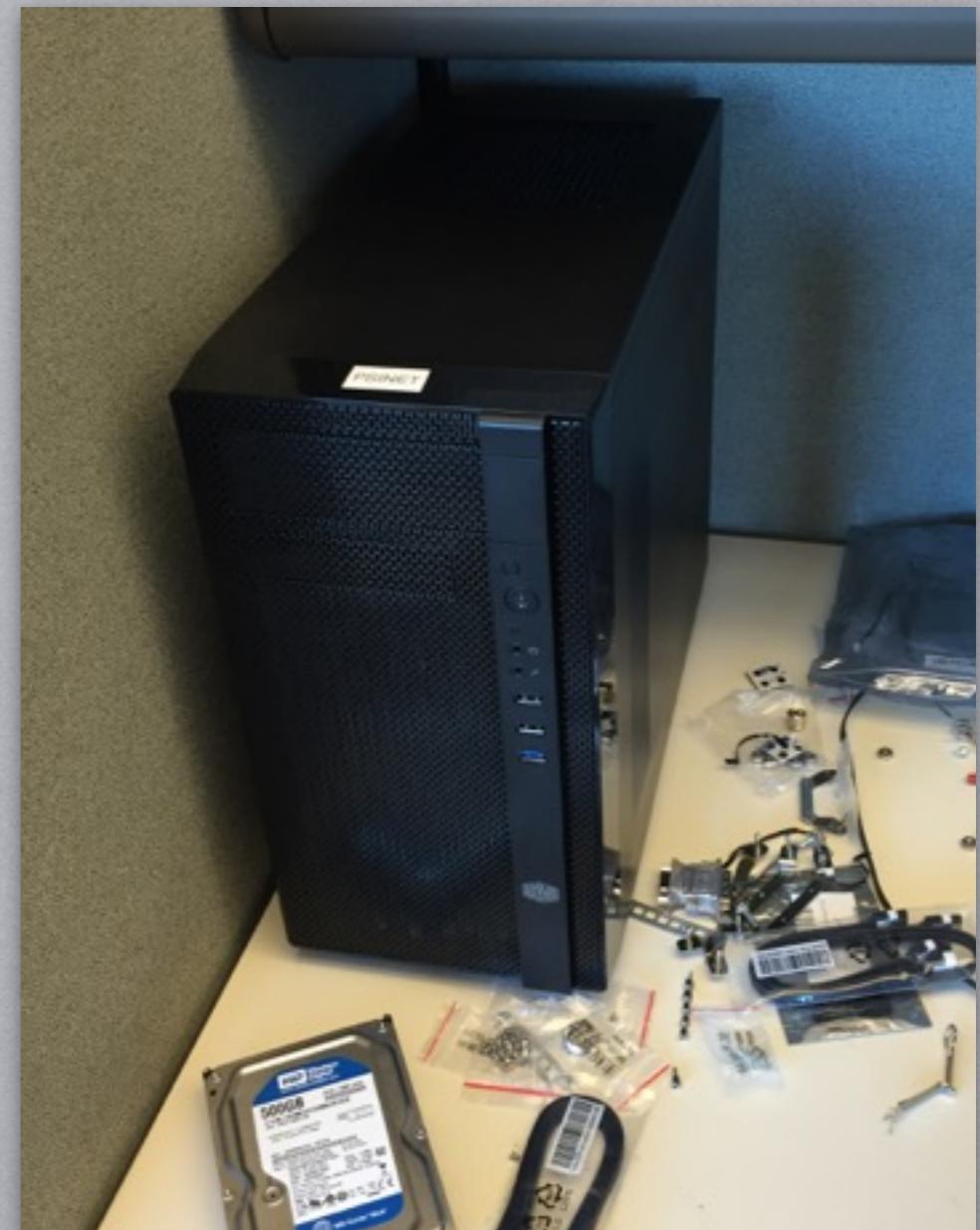
GITHUB ORGANIZATION NEEDS ORGANIZING

- Edu/OS Bronze account with 10 private repos. using 4: psi4, Psi4Plugins, psi4education, JKBuilder
- make your psi4 badge visible
- need to form access teams (i.e., p4-r, p4-rw, & p4p-rw)
- can also form interest teams for mention (e.g., @psi4/core, @psi4/p4edu)
- anyone know why upstream has rw on forks?
- direct access not always needed. in supervisor/ supervised arrangements, latter can clone/fork former's clone/fork

Repository action	Read access teams	Write access teams	Admin access teams	Owners team
Pull (read), push (write), and fork (copy) <i>all</i> repositories in the organization				x
Create and delete any repository in the organization				x
Create and delete the team's repositories			x	x
Change settings for repositories assigned to the admin team			x	x
Transfer repositories into, and out of, the organization account			x	x
Pull from (read) the team's assigned repositories	x	x	x	x
Push to (write) the team's assigned repositories		x	x	x
Fork (copy) the team's assigned repositories	x	x	x	x
Send pull requests from forks of the team's assigned repositories	x	x	x	x
Merge and close pull requests		x	x	x
Open issues	x	x	x	x
Close, reopen, and assign issues		x	x	x
Apply labels and milestones	x	x	x	x
Create and edit releases	x	x	x	x
View draft releases		x	x	x
Edit and delete their own comments on commits, pull requests, and issues	x	x	x	x
Edit and delete anyone's comments on commits, pull requests, and issues		x	x	x
Edit wikis	x	x	x	x
View published releases	x	x	x	x

PSINET

- undisclosed location, GaTech
- RHEL7; Intel C++, Fortran, MKL
- nightly:
 - build linux **binary** from fresh checkout of psi4/psi4 then upload to anaconda.org
 - run all **tests** with binary executable then upload to testboard.org
 - build sphinx **documentation** then upload tgz to psicode.org
 - format GitHub **feed** from git history then upload to psicode.org
- other tasks it could do:
 - regular builds of dependencies (e.g., CheMPS2, dftd3)
 - longer-interval costlier builds (e.g., ang. mom. 8)
 - doxygen documentation
 - anything else community suggests
- **CI!** - TeamCity? want non-cloud sol'n



ANACONDA BINARY

- operational for 8 months
- 800+ downloads, so maybe 80 users
- apart from some flubs on my part, only problems have been glibc too old, which is an expected (and fading) problem
- Mac binary would be trivial, but an optimized math version would need testing of veclib vs. MKL. regular build location also uncertain.
- I've started using binary as the production Psi4 on heterogeneous clusters
- conda in general you should give a try, Mac & Linux, for scientific software. Continuum or the community is paying more attn to scientific software updates than an OS PM
- if anyone has managed to compile Psi4 against Anaconda Python3 with Boost, w/i our build system, please recipe
- should investigate Docker

Click on a badge to see how to embed it in your web page:

Binstar Build No Builds

Binstar 0.3.198

license GNU General Public License v2 or later GPLv2

downloads 858 total

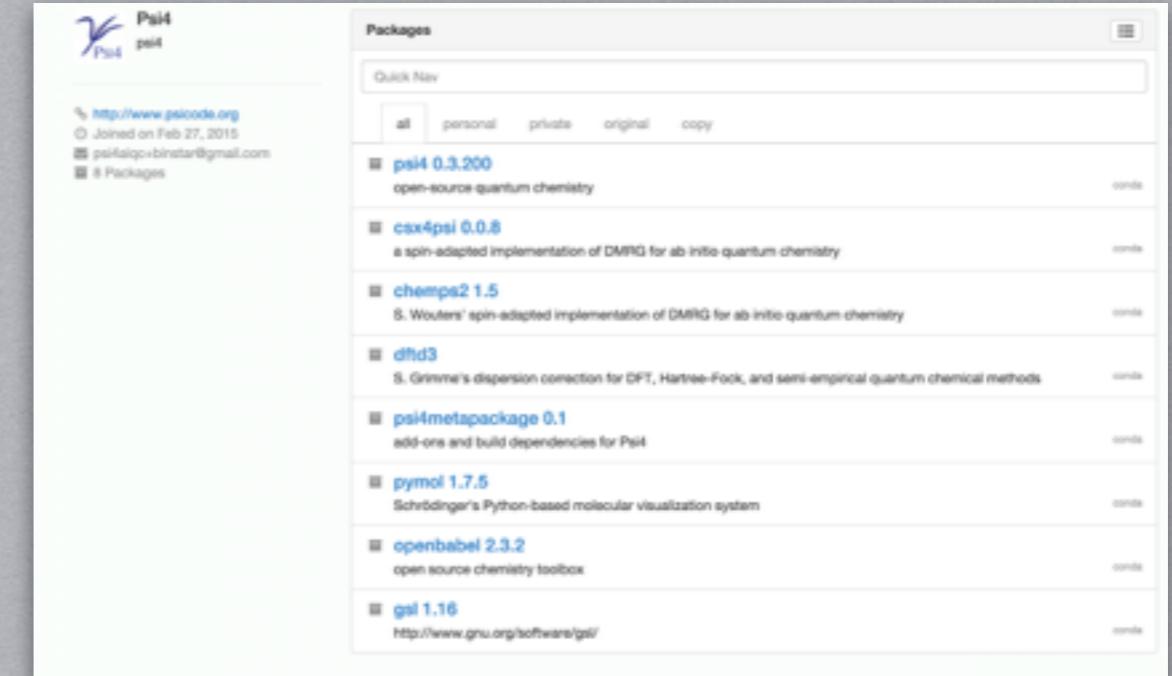
Install with conda

The screenshot shows the anaconda.org website interface for the 'psi4 / psi4' package. At the top, there's a green header bar with the site name, a search bar, and navigation links for About, Pricing, and Help. Below the header, the package name 'psi4 / psi4' is displayed along with its description: 'open-source quantum chemistry'. The main area features a table listing four available builds, each represented by a row with columns for type (conda), size, name, uploaded date, uploaded/built by (psi4), downloads (0), and channels (with a plus sign). The table includes headers for Description, Files, Channels, Metrics, Badges, Builds, and Settings.

	type	size	name	uploaded	uploaded/built by	downloads	channels
0	conda	41.8 MB	linux-64 psi4-0.3.200-py27_gcc67b7b.tar.bz2	10 minutes and a few seconds ago	psi4	0	+ (green)
1	conda	41.8 MB	linux-64 psi4-0.3.198-py27_g75ec49d.tar.bz2	5 days and 10 minutes ago	psi4	21	+ (green)
2	conda	41.8 MB	linux-64 psi4-0.3.198-py27_g30f5e4f.tar.bz2	6 days and 10 minutes ago	psi4	9	+ (green)
3	conda	41.7 MB	linux-64 psi4-0.3.185-py27_g8782e08.tar.bz2	10 days and 10 minutes ago	psi4	15	+ (green)

ANACONDA PACKAGING

- conda packaging encourages modular structure
 - shared library modules (e.g., CheMPS2)
 - executables (e.g., dftd3)
 - Psi4 plugins (e.g., csx4psi)
 - Psi4 itself
 - borrow non-QC dependencies: hdf5, gsl, boost
- maintain Psi4 metapackage
- psinet needs to be doing the packaging for max static-ness but can offer add-on developers:
 - complete control over their codebase since source cloned or ftp'd
 - fast binaries independent of Psi4
 - independent download tracking
 - separation between code and any patches needed by Psi4
- if shared libraries present, need to redistribute MKL libiomp5.so
- would be nice to separate out libint & libderiv so can compile once and for all with insane AM
- eventually want even developers to obtain non-development components from conda
- eventually want to get plugins build able against conda Psi4, if possible



MIRRORS, FORKS, PR, CI, ETC.

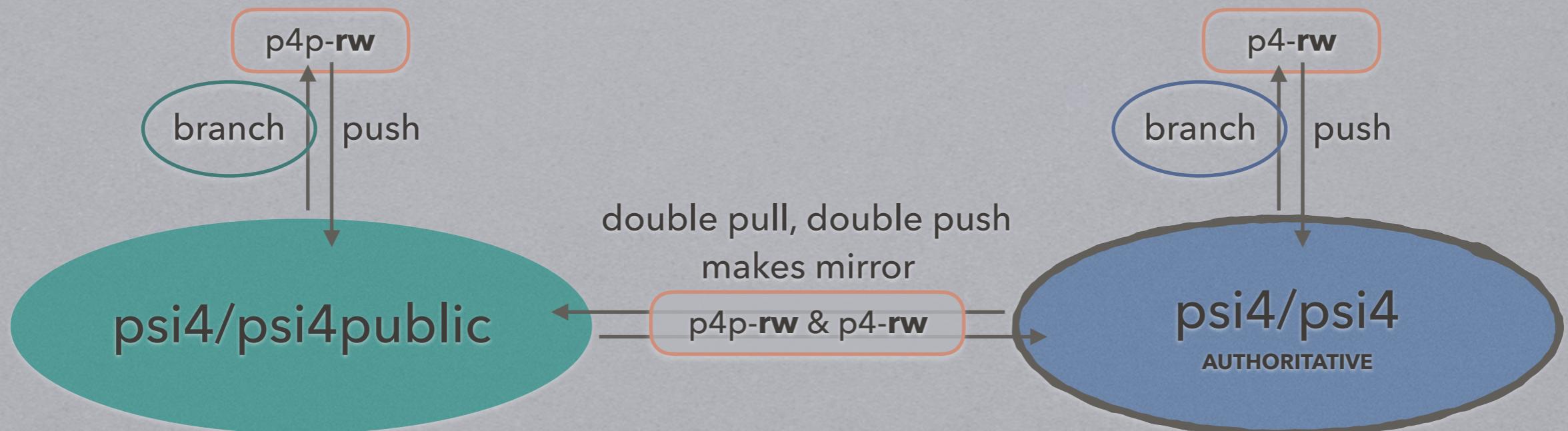
CHECKING CODE INTO Psi4 – LEGEND



MIRRORS, FORKS, PR, CI, ETC.

CHECKING CODE INTO Psi4 – COMMON

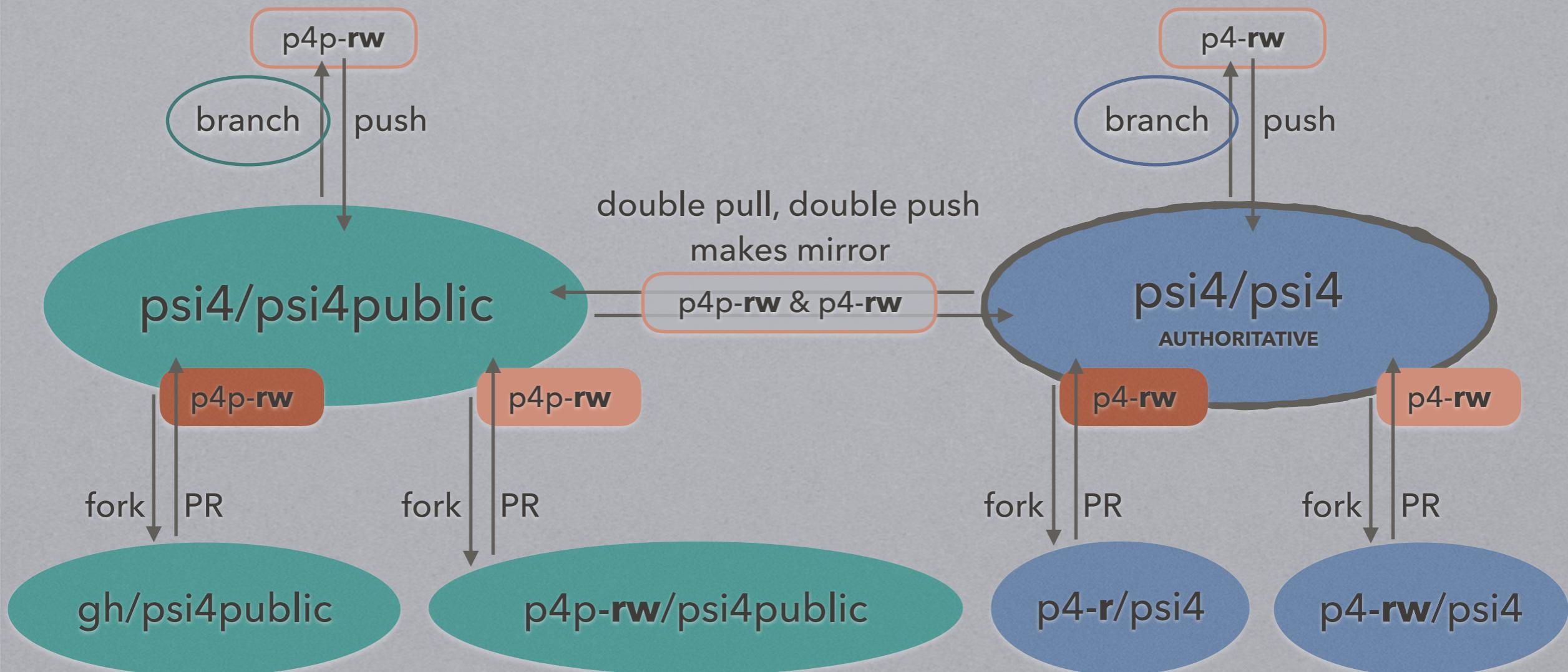
only 2-way paths shown



- HAVE: allow public viewing and private develop

MIRRORS, FORKS, PR, CI, ETC.

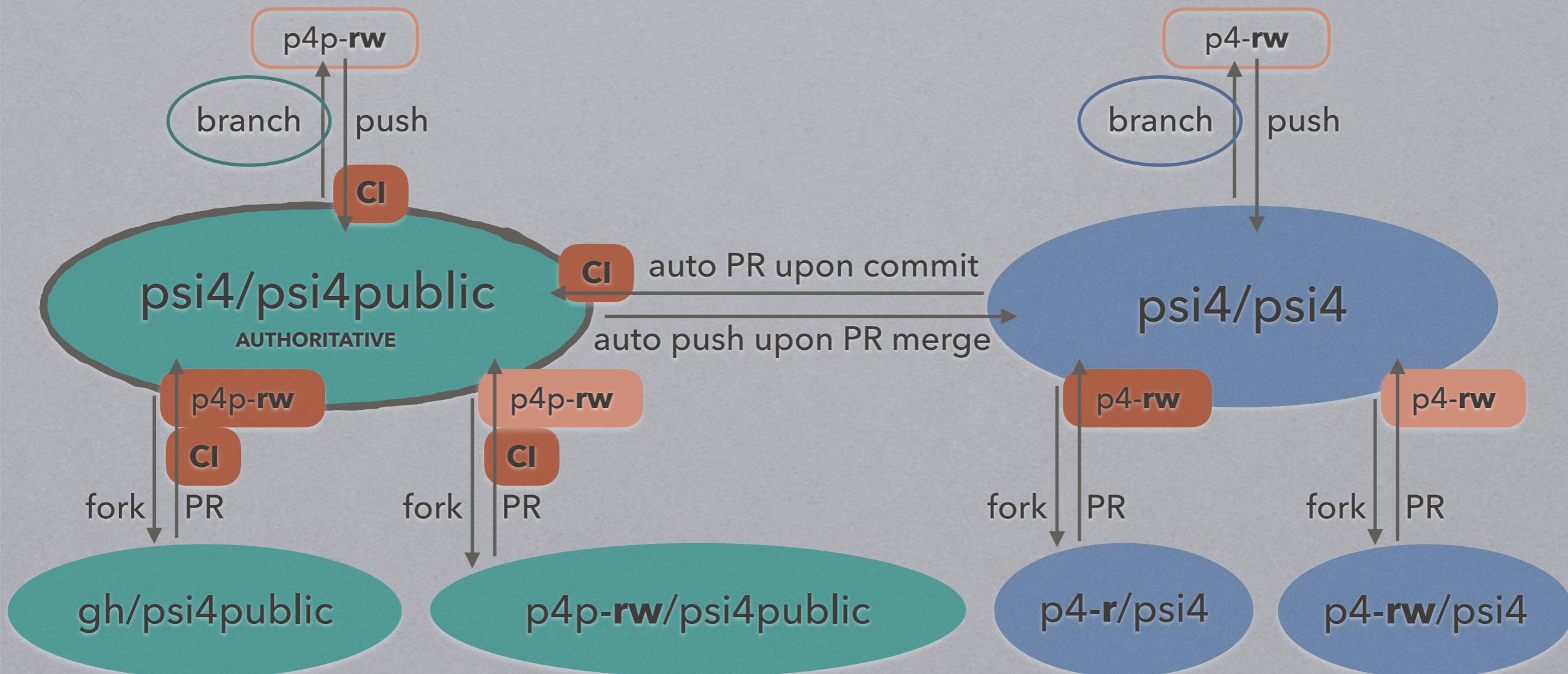
CHECKING CODE INTO Psi4 - NOW



- HAVE: allow non-affiliated people to make contributions

MIRRORS, FORKS, PR, CI, ETC.

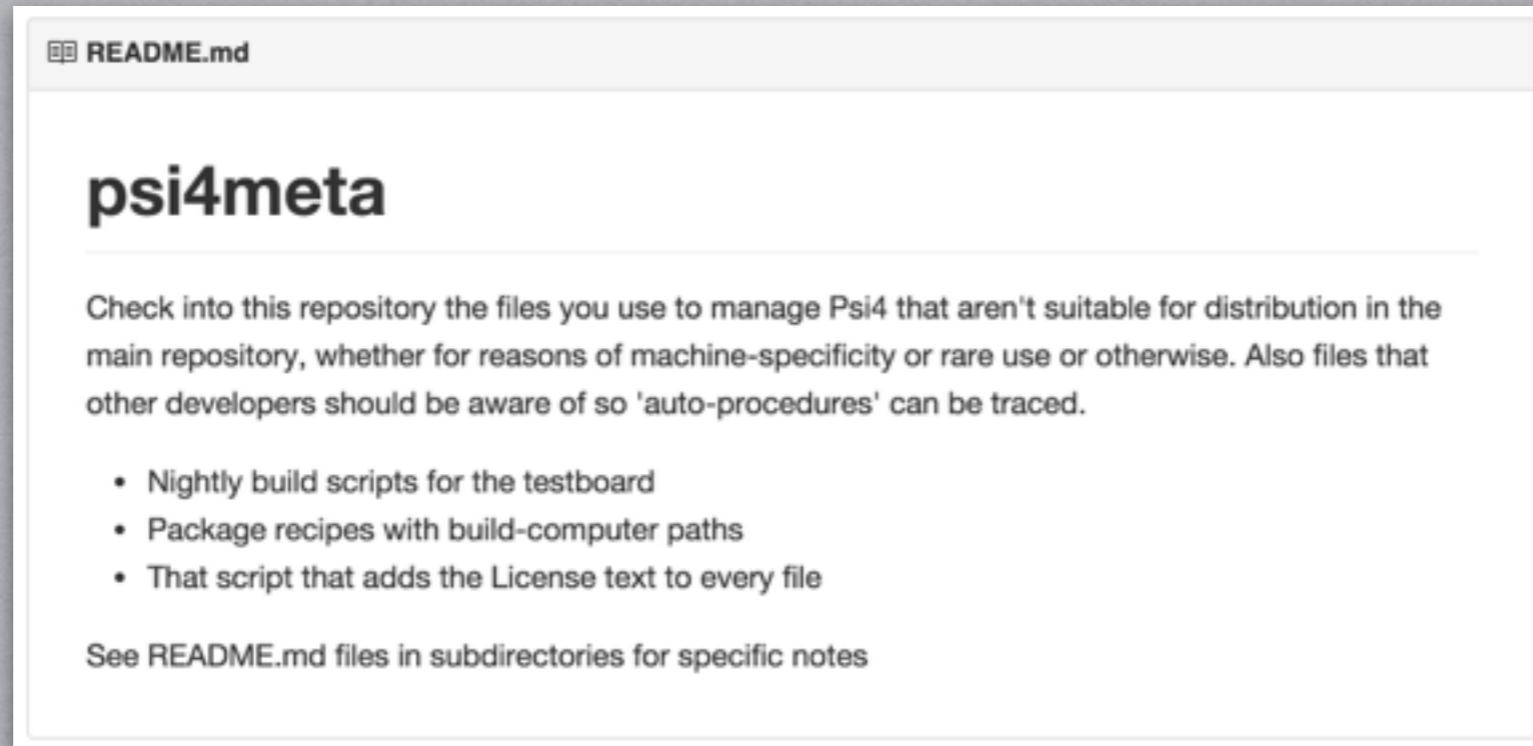
CHECKING CODE INTO PSI4 – SOMEDAY



- WANT: automatic merge-ability and tests-passing checks and priv/pub synch

PSI4META

TOOLS FOR MANAGING PSI4



- GitHub [psi4/psi4meta](#)
- many computer names & paths checked-in. public repo but not so exposed
- stuff anything here that is useful for maintenance but unsuitable for users

PSI4 ON THE WEB

A PLACE FOR EVERYTHING & EVERYTHING IN ITS PLACE

- open software tools are proliferating-choose judiciously or we won't be able to find anything
- though using everything is probably better than issues/plans/ developer guidance residing only in email or gchat threads
- willing in principle to separate out a psi4docs repo to spur livelier docs devel, but full-blown docs build needs PSI4 source and PSI4 exe, so non-trivial
- Where should things go? Need one of these ----->
 - Wiki: acquiring/building docs ONLY, drafts
 - Issues: developer-level problems with the code
 - Meta: management repo
 - PEPs: revive these?
 - Forum:
 - user-level problems with the code or build
 - Discourse wiki post for uniting draft doc & discussion
 - investigate Discourse GitHub integration
 - Slack: is nice, I've used it, but it may be one too many places where plans can reside and not be open
 - WANT: I don't care how many docs pieces (doxygen, plugin, etc.), so long as everything links into *the docs*

Psi4: Open-Source Quantum Chemistry

- Users' Website www.psicode.org
- Downloading and Installing Psi4 <https://github.com/psi4/psi4public/wiki>
- Manual <http://bit.ly/psi4manual> (Moved June 2015!, built nightly from master branch)
- Tutorial <http://psicode.org/psi4manual/master/tutorial.html>
- Forum <http://forum.psicode.org>
- Public Github <https://github.com/psi4/psi4public> (regularly synched from private github master)
- Private Github <https://github.com/psi4/psi4>
- Anaconda <https://anaconda.org/psi4> (binary available for Linux ) [Instructions](#)
- Interested Developers <http://psicode.org/developers.php> (welcome to fork psi4/psi4public or store private branches at psi4/psi4)
- Sample Inputs <http://www.psicode.org/psi4manual/master/testsuite.html> (also in share/psi/samples)
- Download Tarball <http://sourceforge.net/projects/psicode/>
- Build Dashboard <https://testboard.org/cdash/index.php?project=Psi4>

Where-to-post summary:

- How do I? -- StackOverflow!
- I got this error, why? -- StackOverflow!
- I got this error and I'm sure it's a bug -- file an issue!
- I have an idea/request -- file an issue!
- Why do you? -- the mailing list!
- When will you? -- the mailing list!
- You suck and I hate you -- contact us privately at [me@glennbeck.com!](mailto:me@glennbeck.com)
- You're awesome -- aw shucks!

ASTROPY MODEL FOR THE QC COMMUNITY?



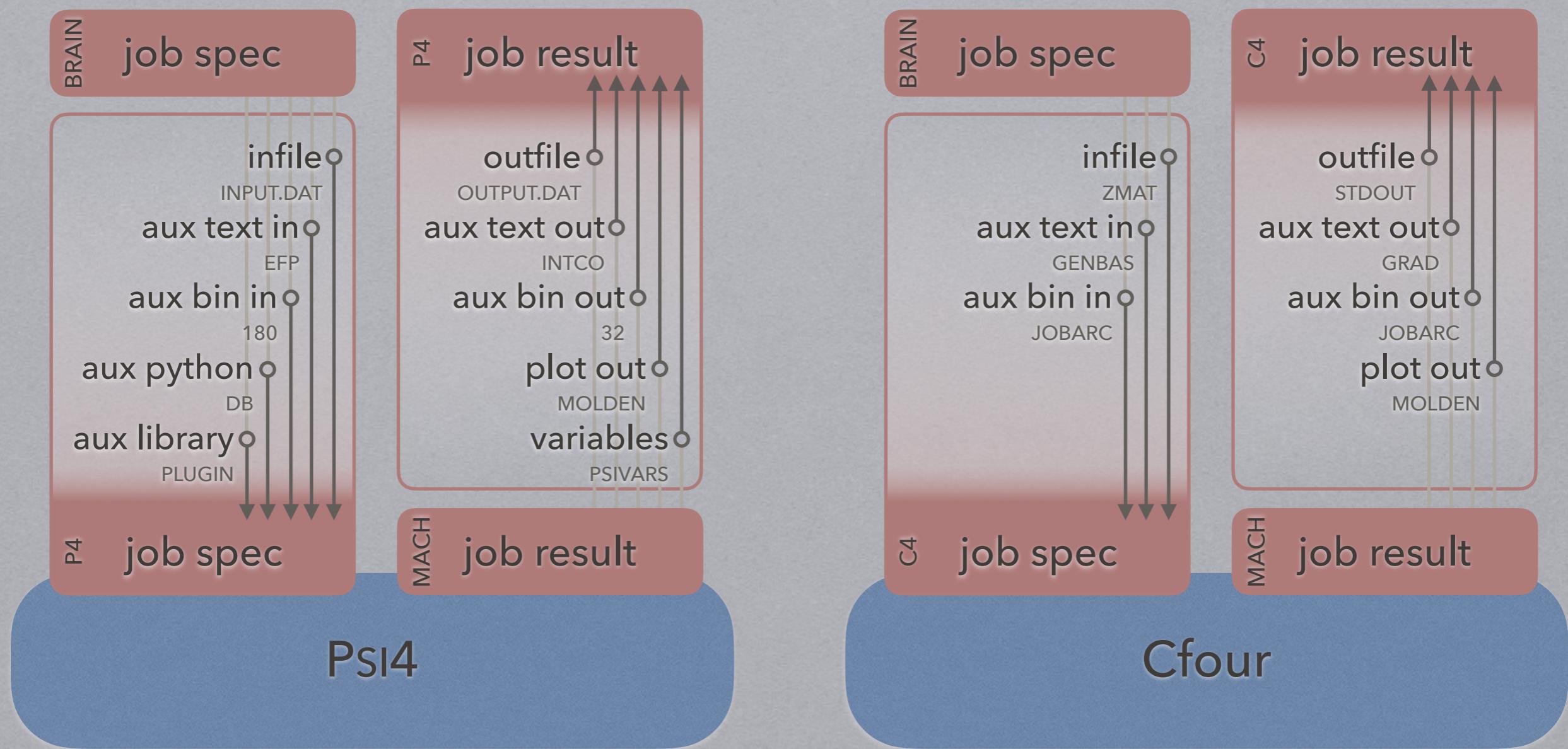
The screenshot shows the official Astropy website at www.astropy.org. The header includes a navigation bar with links to 'astropy', 'About', 'Get Help', 'Contribute', 'Affiliated Packages', and a search bar. Below the header is the Astropy logo and the tagline 'A Community Python Library for Astronomy'. A brief description states: 'The Astropy Project is a community effort to develop a single core package for Astronomy in Python and foster interoperability between Python astronomy packages.' Below this, there are two buttons: 'Current Documentation' (highlighted) and 'Other Docs ▾'. A note indicates 'Current Version: 1.0.6' and a reminder to 'Please remember to [acknowledge](#) the use of Astropy!'. The 'Install Astropy' section features four download links: 'OS X' (selected), 'Linux', 'Windows', and 'Source'. To the right of the main content is a sidebar titled 'Developer'.

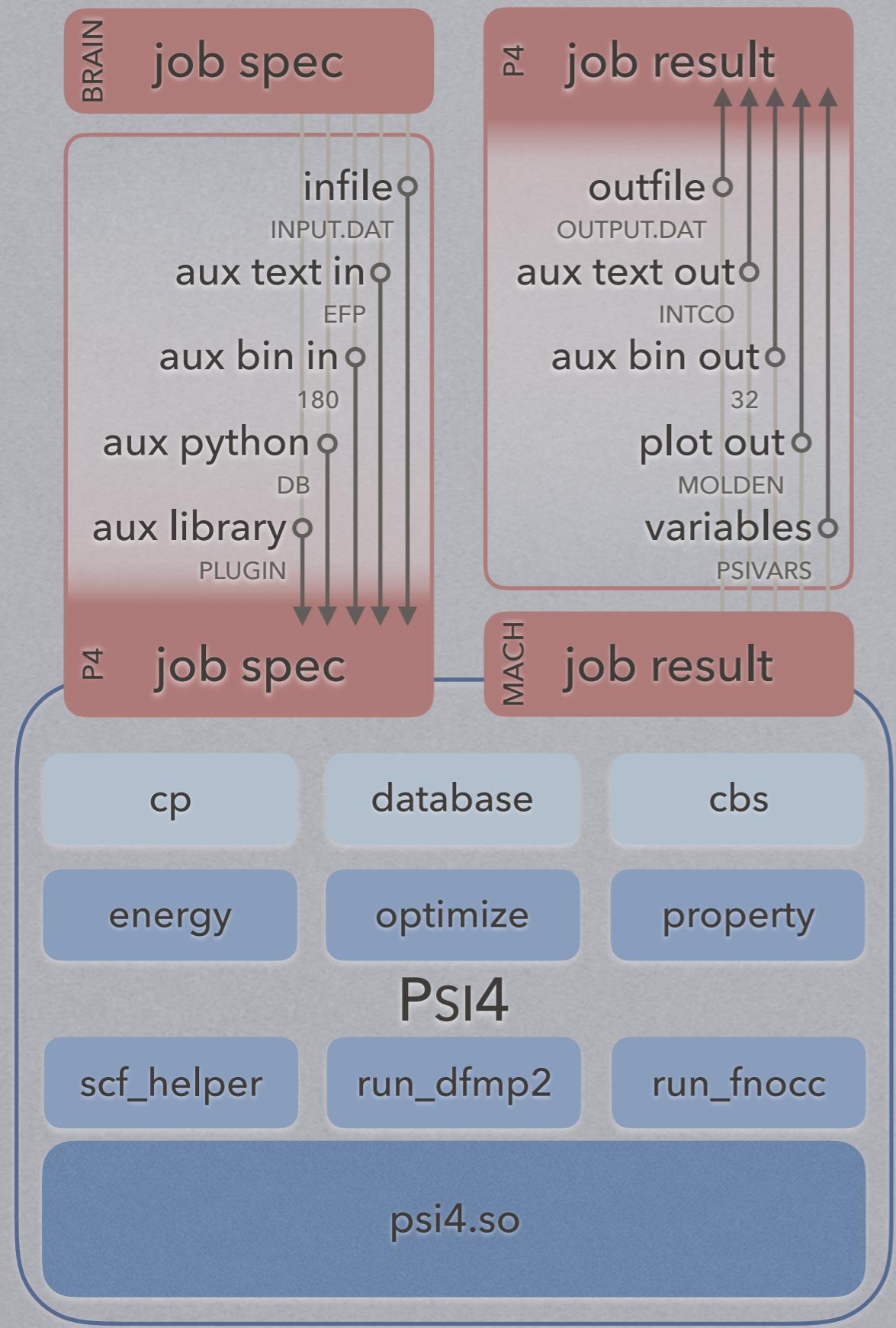
Astropy Project Dashboard

astropy/astropy	build passing	coverage 77%	docs latest
astropy/package-template	build passing	coverage 92%	docs no builds
astropy/astroquery	build passing	coverage 66%	docs latest
astrofrog/reproject	build passing	coverage 98%	docs latest
astrofrog/wcsaxes	build passing	coverage 94%	docs latest
aplpy/aplpy	build failing	coverage 72%	docs latest
pyvirtobs/pyvo	build passing	coverage unknown	docs no builds
eteq/astrophysics	build unknown	coverage unknown	docs no builds

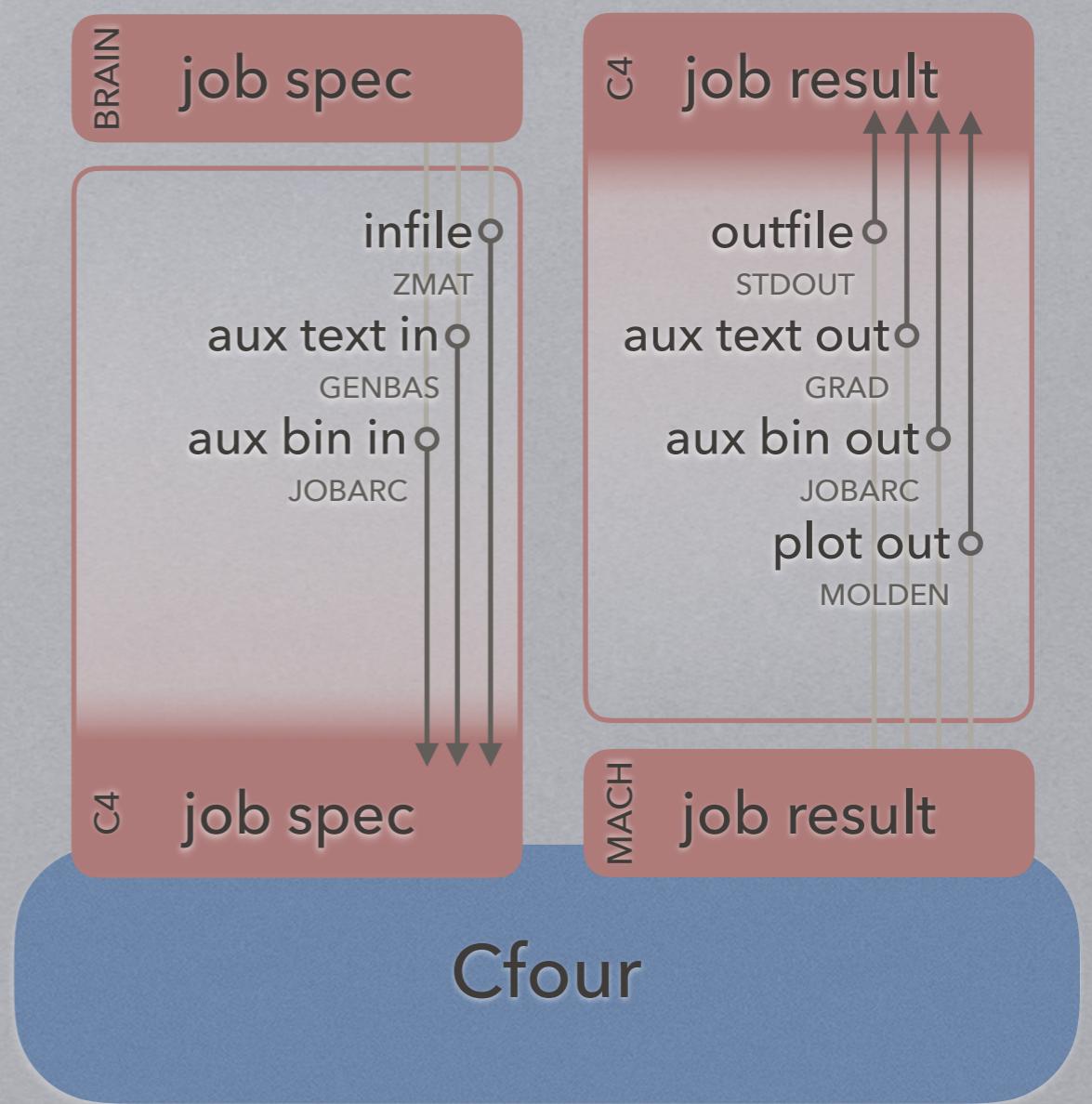
QCDB

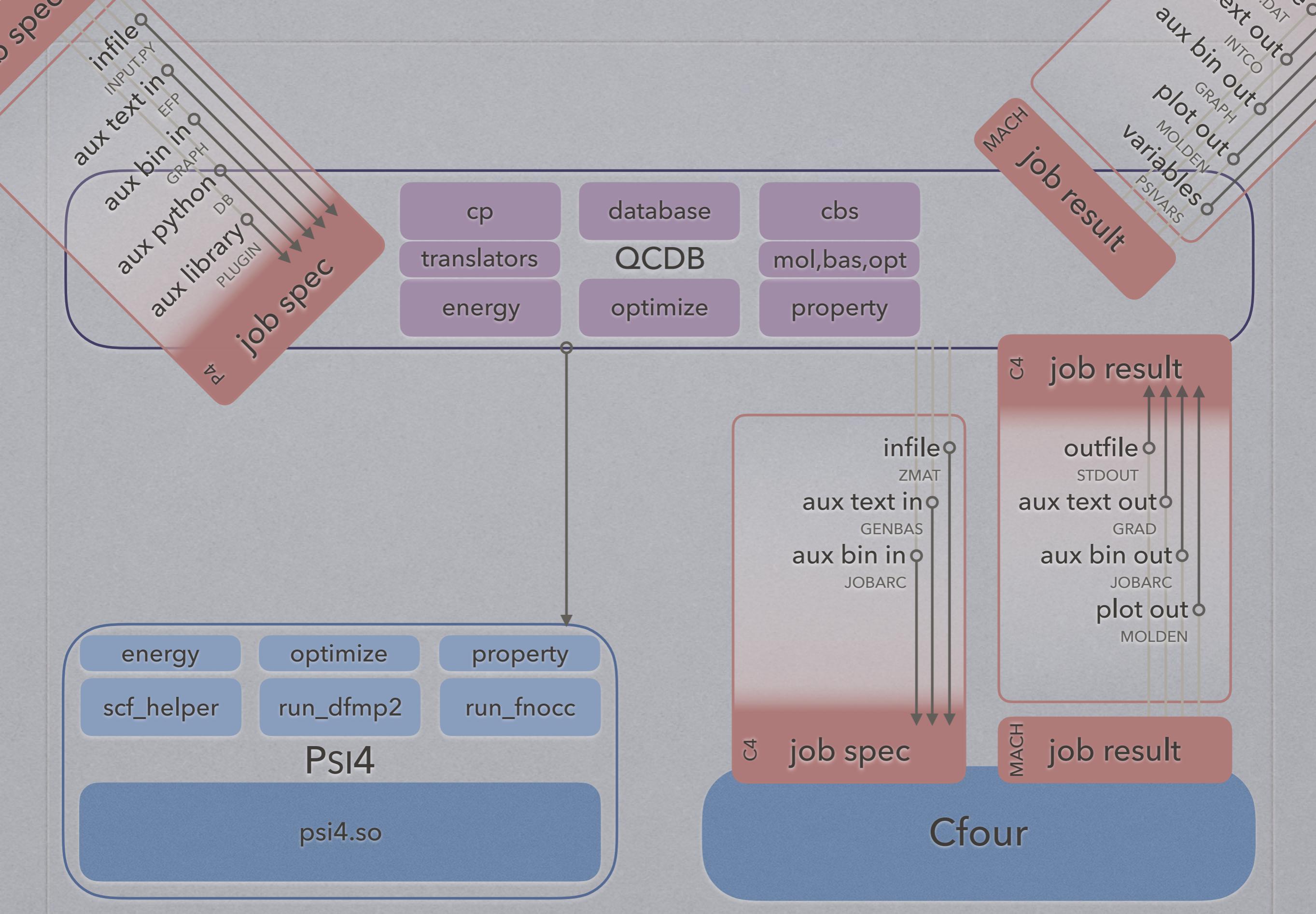
EXE-LEVEL INTEROPERABILITY





- only programmatic access shown
- Boost + Driver yields Python API for QC
- much of driver QC-generic and independent of `psi4.cmd()`, so extract it





QCDB

SOME EASE-OF-USE PRINCIPLES

unified & free formatting

- \s*, case, =, T/1 insensitive
- just b/c obliged to rem. kw doesn't mean should have to rem. formatting, too.

```
set cfour_occupation [[3,1,1,0], [3,0,1,0]]  
OCCUPATION=3-1-1-0/3-0-1-0
```

- let driver handle allotting charge, multiplicity, symm, bohr/ang, cart/zmat, etc. between mol and kw spec. implication that our Mol must subsume capabilities of other programs' Mol

directive & keyword translation

- many user knobs have common purpose

```
set scf d_convergence 5e-6  
SCF_CONV=6
```
- can interject complicated method specification or best practices.

```
energy('c4-ccsd(t)')  
CALC_LEVEL=CCSD[T],CC_PROG=ECC,  
DERIV_LEVEL=0
```

- interpretations can always be turned off

permeable transition

- hard-earned knowledge of program idiosyncrasies can still be put to use
- wrap input bodily: `cfour {...}`
- exotic calcs run, even if output extraction or post-processing not ready.
- use all native kw names in combo with unified mol, bas, kw formatting. best of spec. ease and precision.

```
set cfour_lineq_conv=12
```

RETRIEVING ‘WHAT WAS DONE’

- QC codes satisfactory in conveying to the code through options what we want done. this is by necessity to get #s.
- a certain input file with a certain QCprog version is deterministic, yes (ok, let's resist listing ways that can go wrong), but it's still a rotten state variable for 'what was run' because
 - keywords beget keywords so what was default is no more
 - keywords used as signals by code so values change in course of calc. esp. in sophisticated procedures that try many things to get answer
 - keyword changes may not even be conveyed to the Options object, esp. those whose value (or even default) depends on several others
- so any single options snapshot could well be insufficient. what can be used instead?
 - output file. our primary strategy is to survey the output file and see if the text looks about right
 - Psi4 at least makes Options status available programmatically. helps but incomplete
 - Psi4 does rather better job than other QCprogs in that can access programmatically the key quantities through PsiVars and PsiMats and judge that if they were set, that calc took place. but these, too, can be gamed and sometimes reflect options, not code.
- frankly, Gaussian is who actually attempts to record program flow with the cryptic link and options section
- we use terms sloppily. does 'MP2' mean *the* MP2 or MP2(fc), DF-MP2, SCS-MP2, or anything in the MP2 family?
- solutions
 - deep in the code near the equations, attach values to controlled vocabulary/ontology node that defines it. along the lines of psivars.
 - negotiate keywords btwn user, driver, module defaults, module complex defaults so that a complete state of module operation stored. store for every module/box. this esp. important as new players of qcdb-level vs. qcprog-specific options enter the negotiation.
 - encourage forking/new func/new box/whatever if change to code changes meaning of result variable
- WANT: the program should report back linked and programmatic access to exactly what it did
- would you confidently undertake to reproduce to 6 decimal places in psi4 a number from an unfamiliar QCprog given only its input/output?

THOUGHTS ON MODULES

- boxes are starting to accumulate. if it can be shown that info from immediate parent is sufficient for box to do its task, I approve of the module system over driver scheme.
- opportunities for multi-layer embarrassing parallelism. If box can be either of fully initialized by graph node or commence to build its own boxes to do its tasks, I approve of graph-based restarts.
- puts internal code and contributors code on same footing, though former will get run b/c we set the module defaults
- puts proper QC code and post-processing on same footing, which can't but help evenhanded data management
- may help 'what was done' problem with graph of local, negotiated options snapshots
- may help 'what was done' if encourage small new box for every method variant
- helps form succeed-at-all-costs routines a la Gaussian

INTEROPERABILITY

PLANNING FOR MANY LEVELS

- silo-input level: convey the same instructions to different programs
 - keywords, directives, molecules all boil down to keywords
 - sometimes hampered by type (e.g., discrete conv=5)
 - explicit specification of defaults
- array-level: sharing integrals, densities, amplitudes
- silo-output level
 - does 'MP2' mean *the* MP2 or MP2(fc), DF-MP2, SCS-MP2
 - many programs make determination uncertain from output alone
 - WANT: the program should report back exactly what it did
- WANT: always plan as if we're only a small library/cog of something bigger later

ADD-ON IDEALS

- after much wrangling of CMake and CheMPS2, some thoughts good practices
- let us use a fixed pattern in setup options, CMake variables, and preprocessor names for all Add-ons
- most flexible arrangement is to bully Psi4's and Add-on's CMake builds into allowing 3 routes:
 - detecting pre-built dynamic Add-on for headers and link-stage (needed so Psi4 condapkg can use Add-on condapkg). this requires writing a FindAddon.cmake module- not too hard.
 - building-on-demand dynamic Add-on (needed to prototype all the CMake variable exchanges that the parent and external project are going to have to make). this requires writing a ConfigAddon.cmake module with an extensive ExternalProject_Add command.
 - building-on-demand static Add-on (not needed for conda scheme, but this is what most everyone actually uses b/c all get bundled into the executable).
- you and Add-on developer need to be on good terms and starting from an orderly and conforming CMake because you're going to be exporting a lot of variables to make sure that Add-on is detecting same dependencies as Psi4, that Add-on is using same math libraries (FindBlas won't cut it), that conda package is generic and maximally static, and that Add-on installs where you need it to.
- if all this sounds very complicated even for a clean and Debian-fearing add-on, you're right. imagine how ugly it could get separating out an add-on whose build has become enmeshed in Psi4's build system.
- WANT: separate projects to remain separable, not absorbed. possible implication that I favor add-on detection/ExternalProject over git submodules and mega-executable
- if you read too many CMake docs & guides, you'll notice that CMake isn't designed to rigorously mix one from-scratch build and many external-project builds. they'd rather you did a Psi4-up-to-library CMake project, then a big metaproject that found dependencies Psi4, Boost, CheMPS2, PCMSolver, etc. and just did a big link. I'm not about to do a fourth Psi4 CMake version for idealism, but worth noting.

Imposing your will upon CMake

- Relevant **setup Options:**

--chemps2 {on,off}
--chemps2-dir PATH

Enable CheMPS2 external project [default: on]
The install directory for a pre-built CheMPS2.
'lib/chemps2.so' and 'include/chemps2/DMRG.h' should
be in this directory. If this is left blank cmake will
build one for you. (default: None)

- Relevant **cmake Options:**

-DENABLE_CHEMPS2=ON
-DCHEMPS2_ROOT=PATH

- enable CheMPS2 (OFF to disable)
- Preferred CheMPS2 directory

- Relevant Preprocessor Options:

ENABLE_CHEMPS2

- set when CheMPS2 is enabled

DEVELOPERS LINE-UP AVAILABLE



Rob Parrish
Stanford



Rollin King
Bethel



Daniel Smith
Auburn



David Sherrill
GaTech



Lori Burns
GaTech



Ryan Richard
GaTech



Francesco Evangelista
Emory



Eugene DePrince
FSU



Fritz Schaefer
UGA



Justin Turney
UGA



Daniel Crawford
VaTech



Ed Valeev
VaTech



Andy Simmonett
NIH



Alex Sokolov
Princeton



Ed Hohenstein
CCNY



Roberto Di Remigio
Tromso



Ugur Bozkaya
Ataturk

- criterion: 50 commits + 2 * Ed, arranged W to E
- not trying to filter contributions, just stating this pic is available for adaptation