

Abstract

Simplifying Multilevel Quantum Chemistry Procedures through Psi4 and QCARCHIVE: The Psi4 quantum chemistry (QC) program is reworking its outer Python layer to facilitate high-throughput computing. For users, this allows naturally parallel procedures such as composite methods or many-body routines to run in parallel with minimal changes to the input. Central to this effort is interfacing with The Molecular Sciences Software Institute's Quantum Chemistry Archive (QCA) project to provide database storage and query and promote standard interfaces, QCSchema, for communication between software projects in the field. Capabilities to call Psi4 and other QC programs through increasingly uniform input suitable for software generation will also be presented.

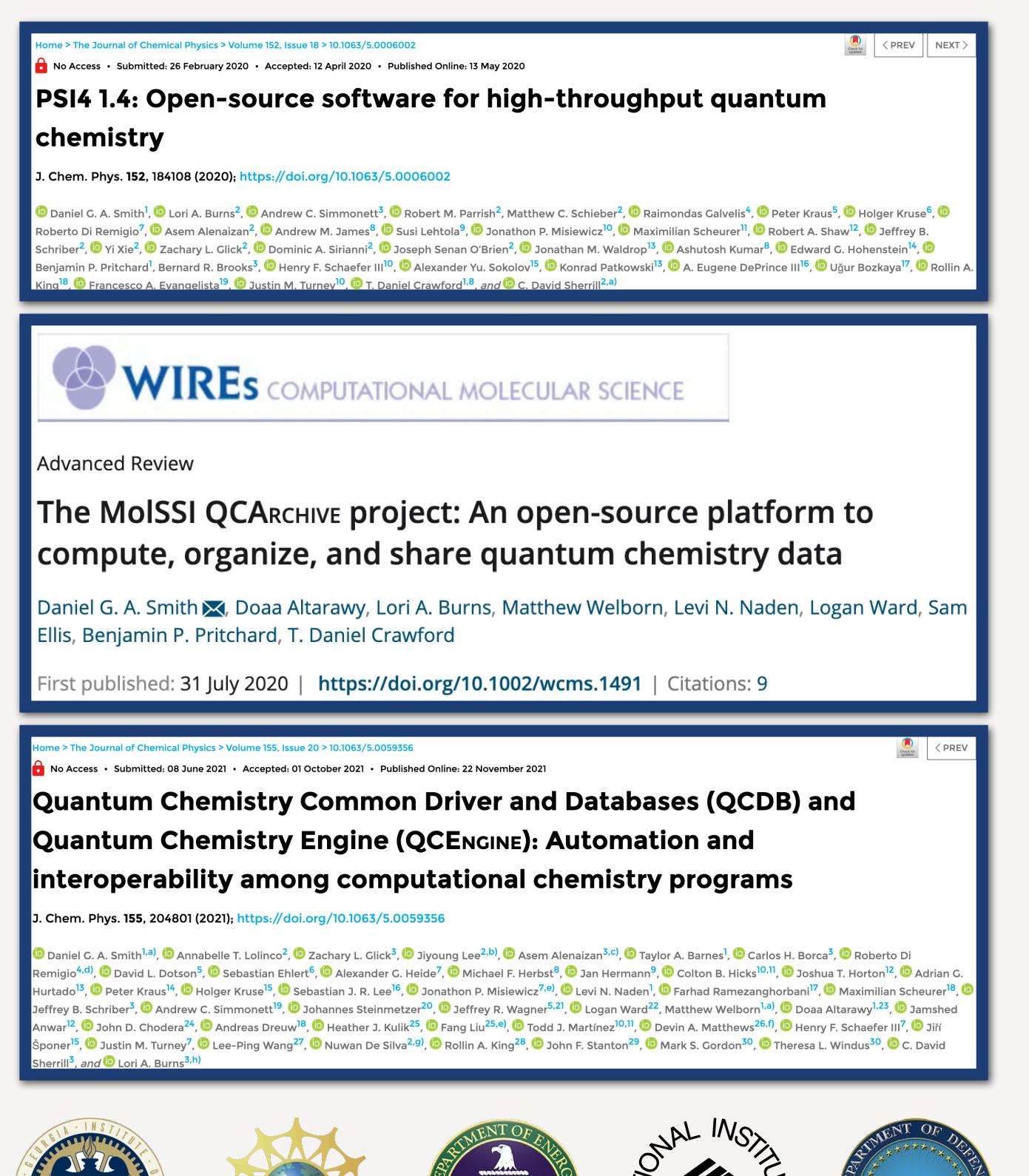
Demo

```
import psi4
from qcfractal import FractalServer
server = FractalServer()
client = server.client()

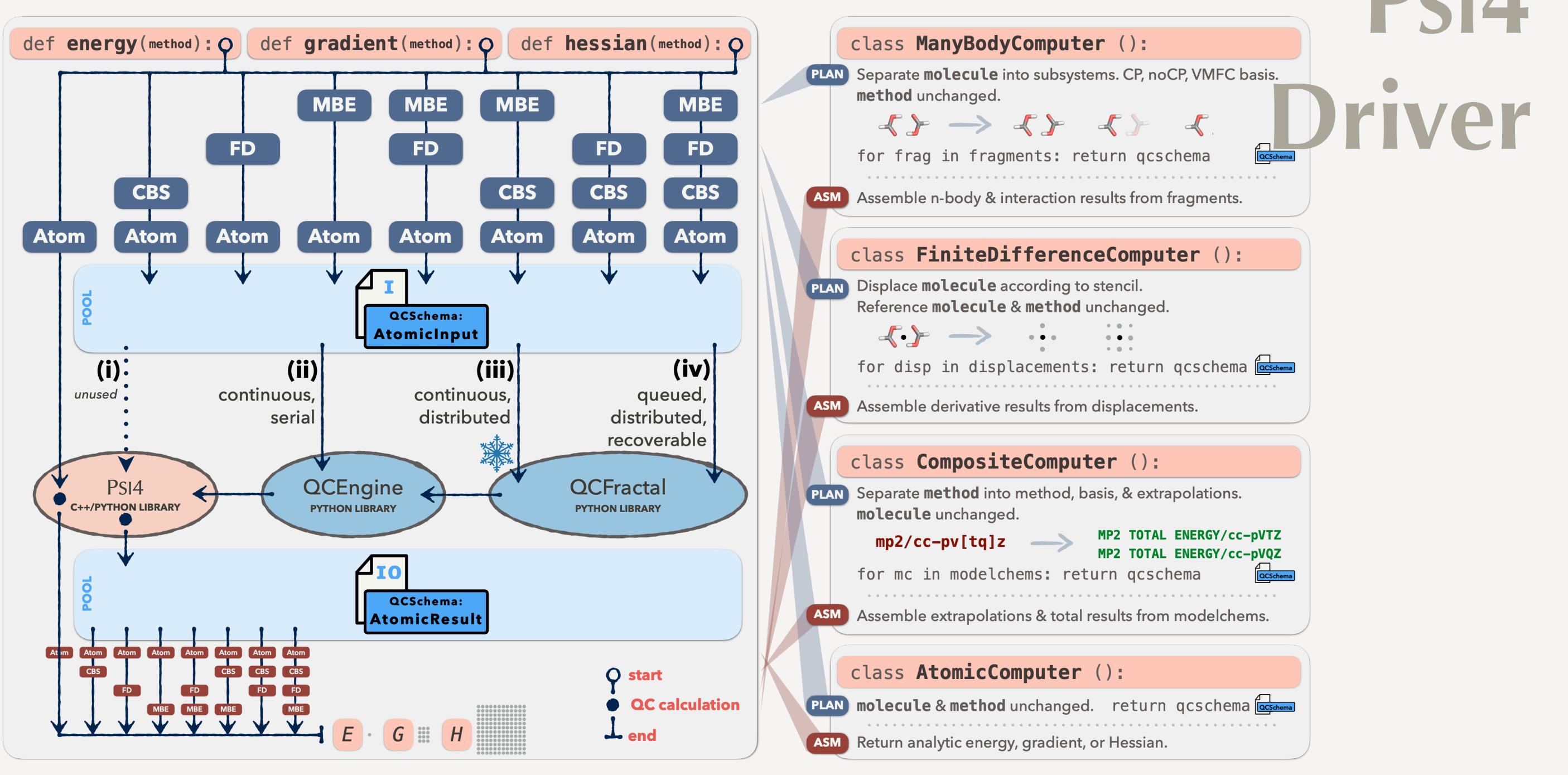
plan = psi4.energy(
    "MP2/cc-pVTZ+d:CCSD(T)/cc-pVQZ",
    bsse_type="cp",
    return_plan=True,
    molecule=water
)
plan.compute(client)

import qcfractal
# read run file after jobs complete for final processing
server = FractalServer()
client = server.client()
print(qcfractal.get_results(client))
# print(qcfractal.get_return(result)) # CCSD(T).energy
plan = psi4.energy("CCSD(T)/cc-pVQZ",
    return_plan=True,
    molecule=water
)
plan.compute(client) # freeel calc in database
qcsk = plan.get_results(client)
print(qcsk.return_result) # CCSD(T).energy
```

Publications

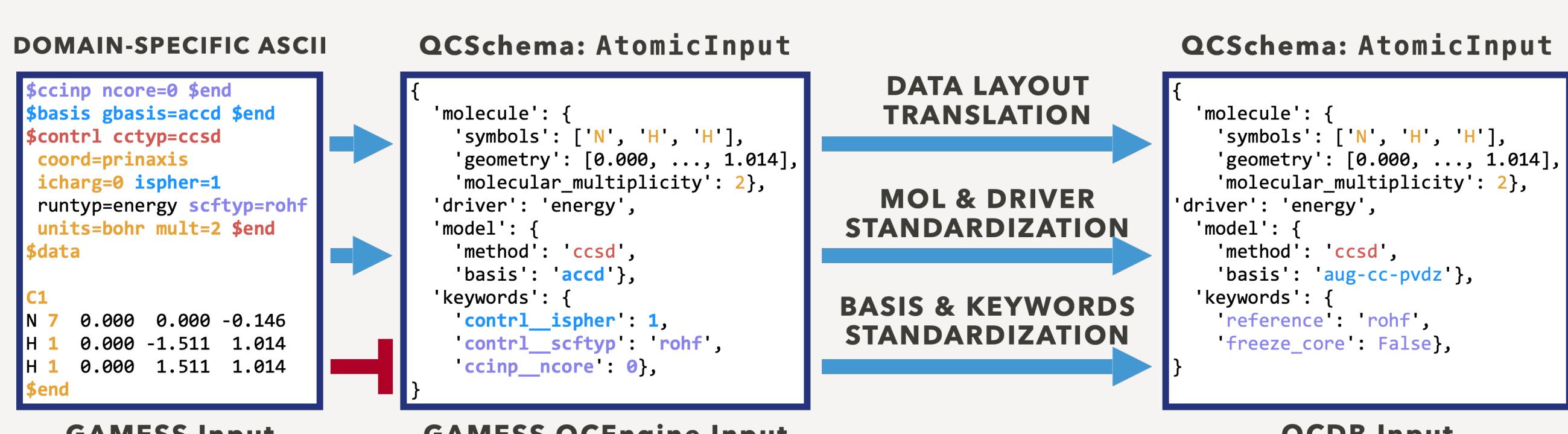


Psi4 Driver



QCDB

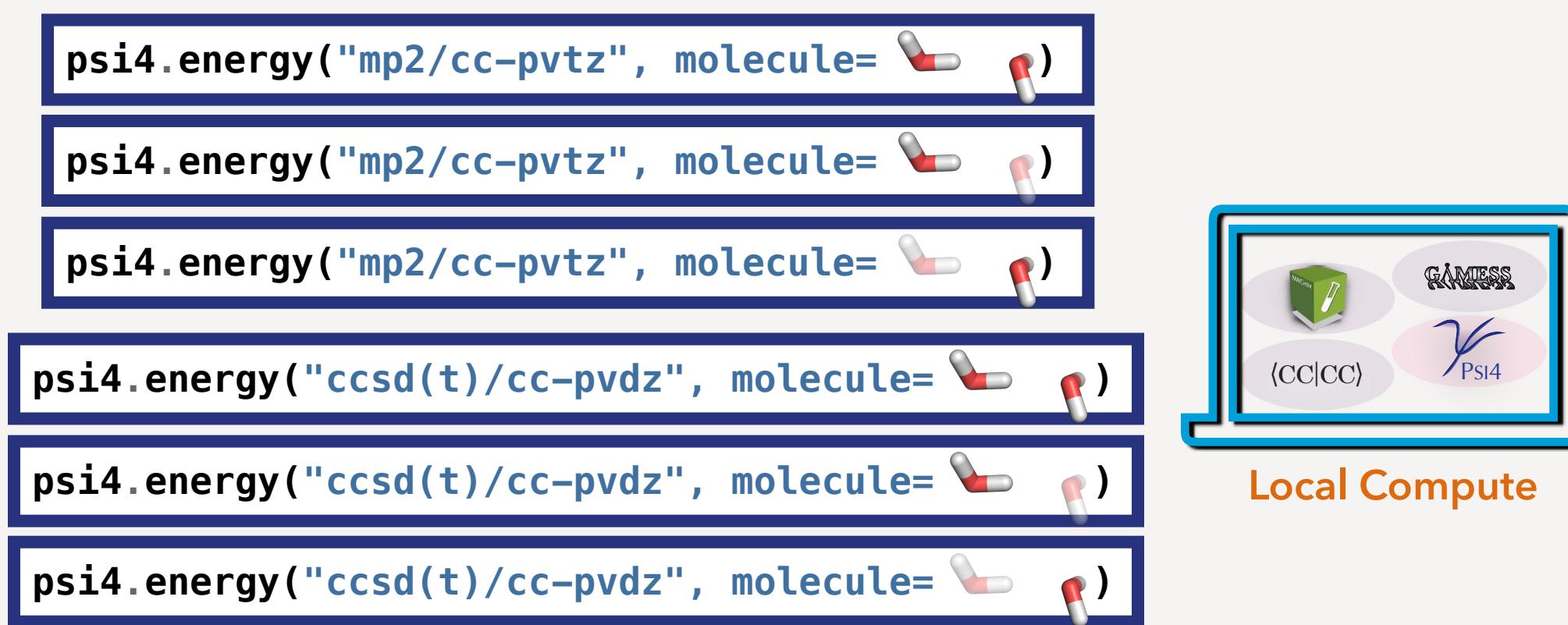
- LAYOUT JSON Schema defines a data layout and some minimal type and count checking.
- CONVENTIONS JSON Schema can't check conventions you rely upon like AU units, CCA ordering, center-of-mass positioning
- QCDB UNIFIED CONTROL & KEYWORD TRANSLATION allows users to focus on scientific choices, not DSL.
- LOWER BARRIERS to using a variety of codes, many with unique features.



Case Study

COMPOSITE & BSSE BY HAND

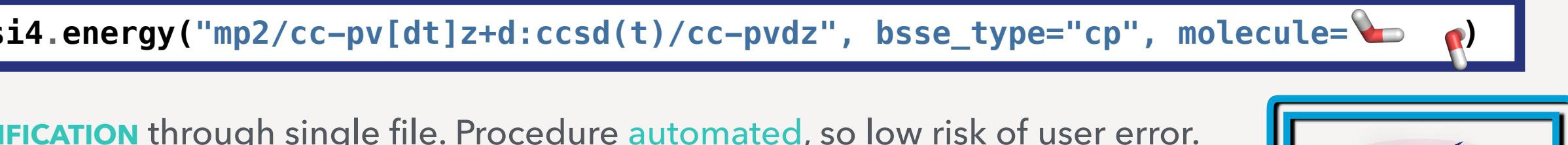
flexible but error-prone



- SPECIFICATION through multiple files of slight variation. Procedure **manual**, so high risk of user error.
- RATE-LIMITED by single longest calc since can run **parallel in queue**.
- RETRIEVAL from **filesystem** if named systematically.
- FLEXIBILITY to mix QC programs to access more or **more efficient** methods since separate input files.

Psi4 DRIVER, INTERNAL

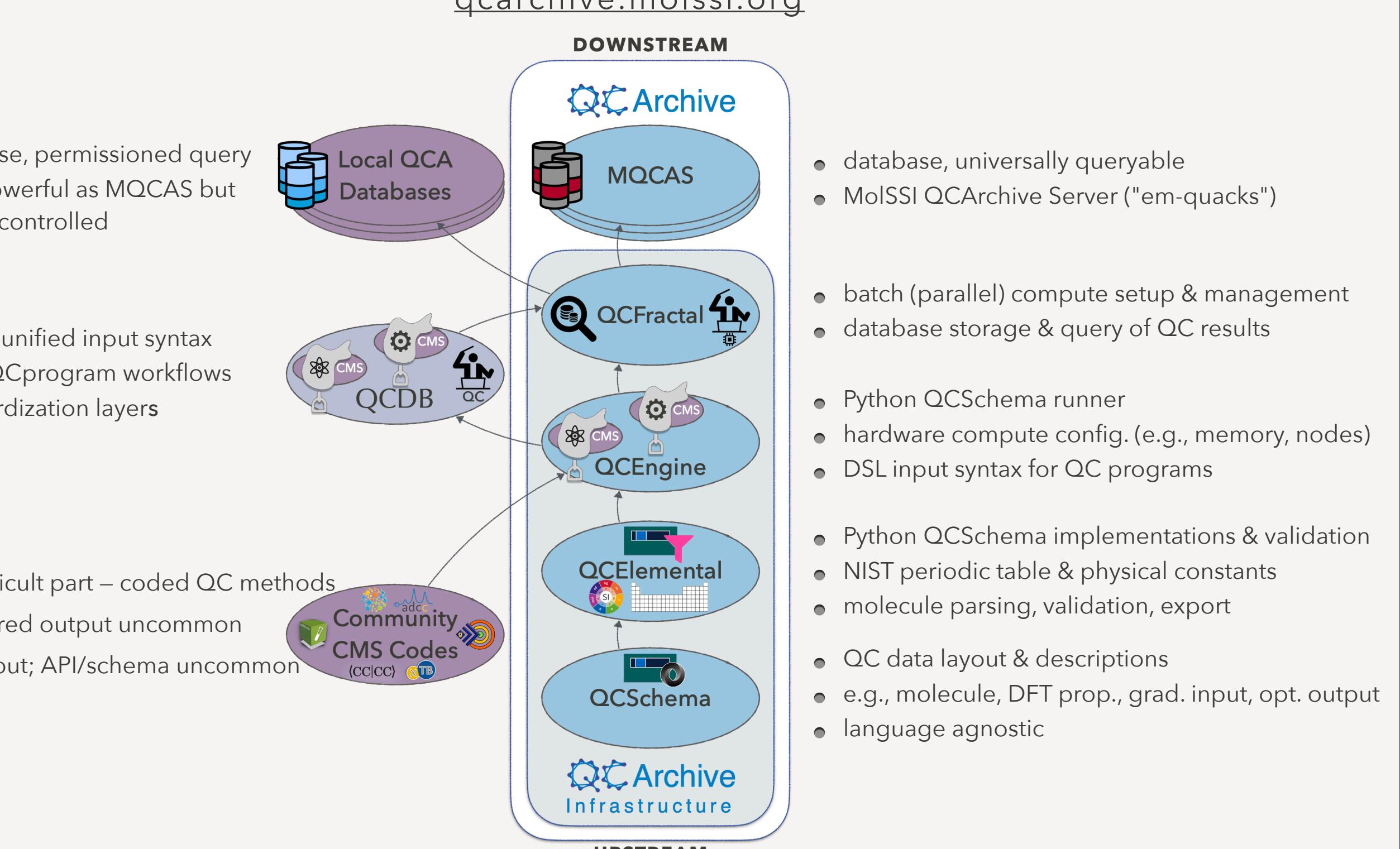
automated and extensible



- SPECIFICATION through single file. Procedure **automated**, so low risk of user error.
- RATE-LIMITED by sum of all calcs since run **sequentially**.
- RETRIEVAL from filesystem difficult since many calcs in aggregated outfile.
- FLEXIBILITY limited to Psi4 only.

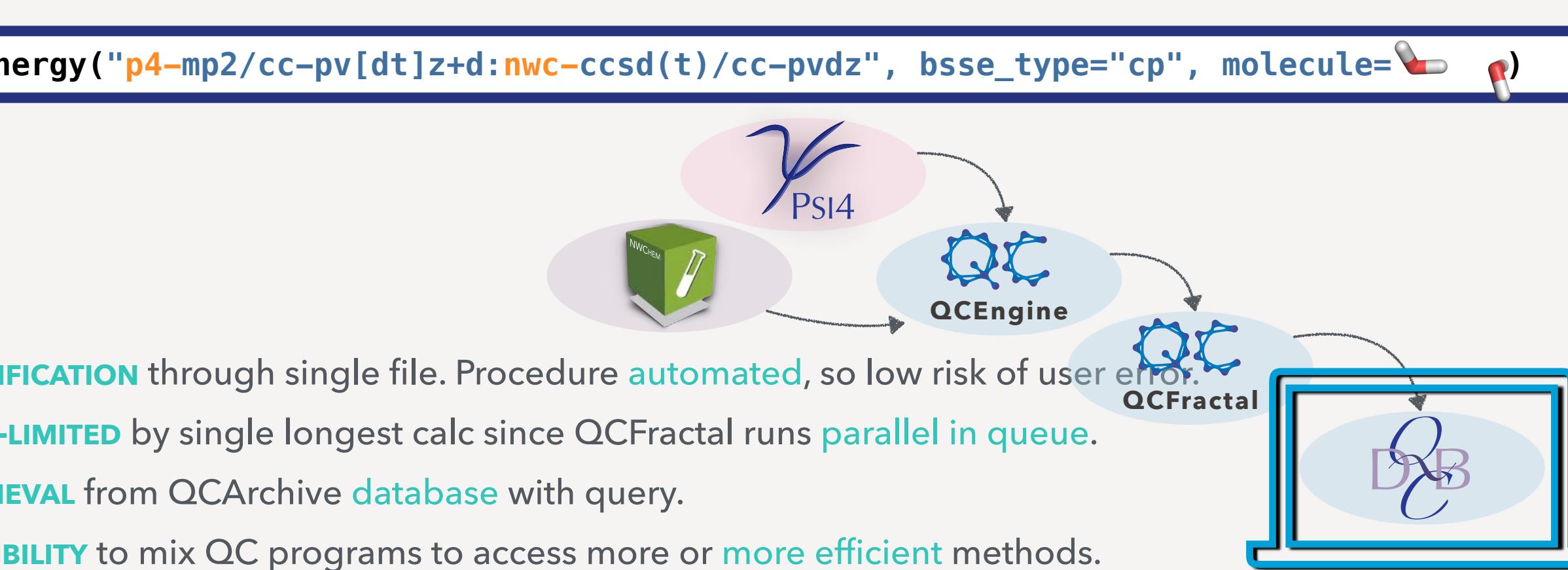
QCARCHIVE STACK

qcarchive.molssi.org



QCDB DISTRIBUTED DRIVER, QCARCHIVE

distributed, searchable, error-resistant, and program-generalized



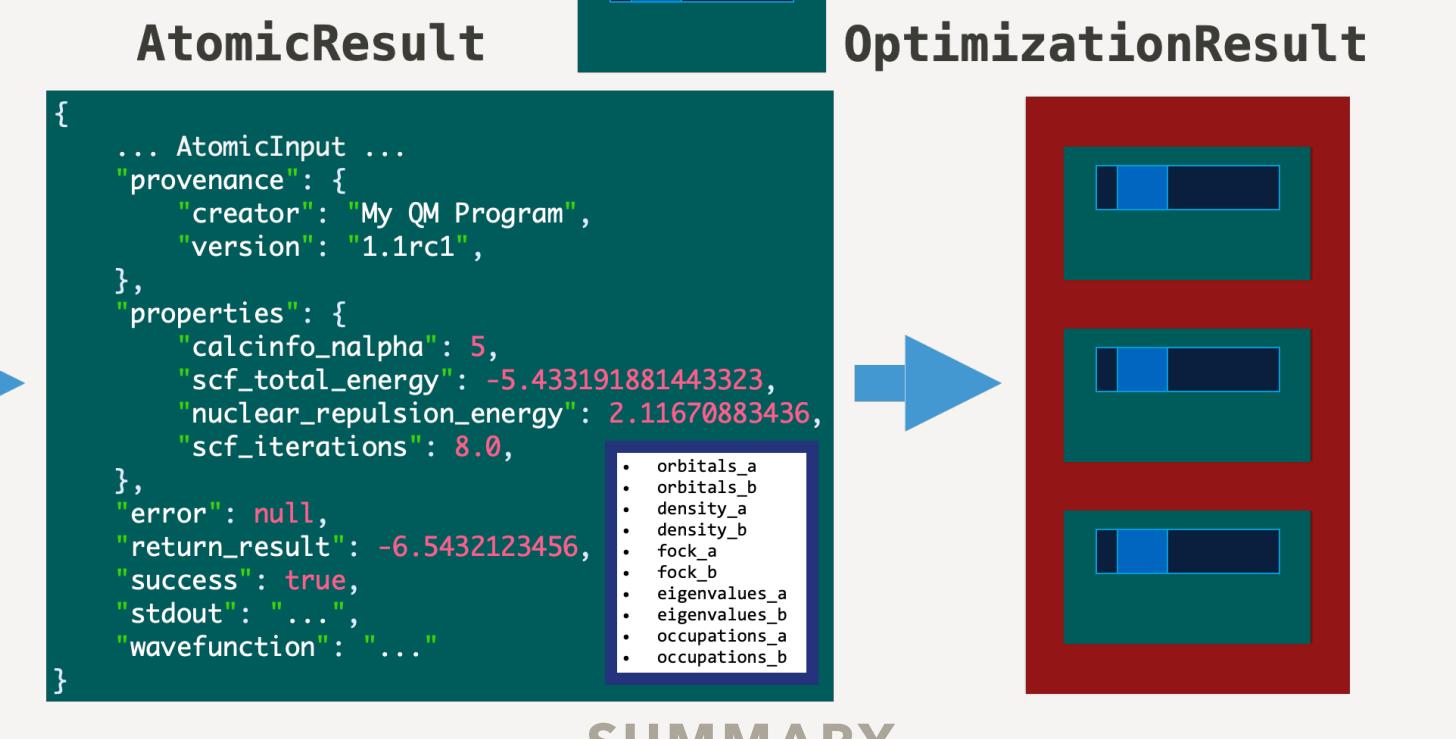
QCSCHEMA

AtomicInput

```
{ ... AtomicInput ...
  provence: {
    creator: "My QC Program",
    version: "1.1rc1",
    properties: {
      calcinfo_nalpha: 5,
      scf_total_energy: -5.433918844532,
      multistep_reputation_energy: -2.1670883436,
      iterations: 8,
    },
    error: null,
    return_result: -6.5432123456,
    success: true,
    stdout: "...",
    wavefunction: "...",
  },
  molecule: {
    geometry: [0, 0, 0, 0, 0, 1],
    symbols: ["He", "He"],
    ...
  },
  driver: "energy",
  model: {
    method: "CCSD(T)",
    basis: "aug-cc-pVQZ",
    ...
  },
  keywords: []
}
```

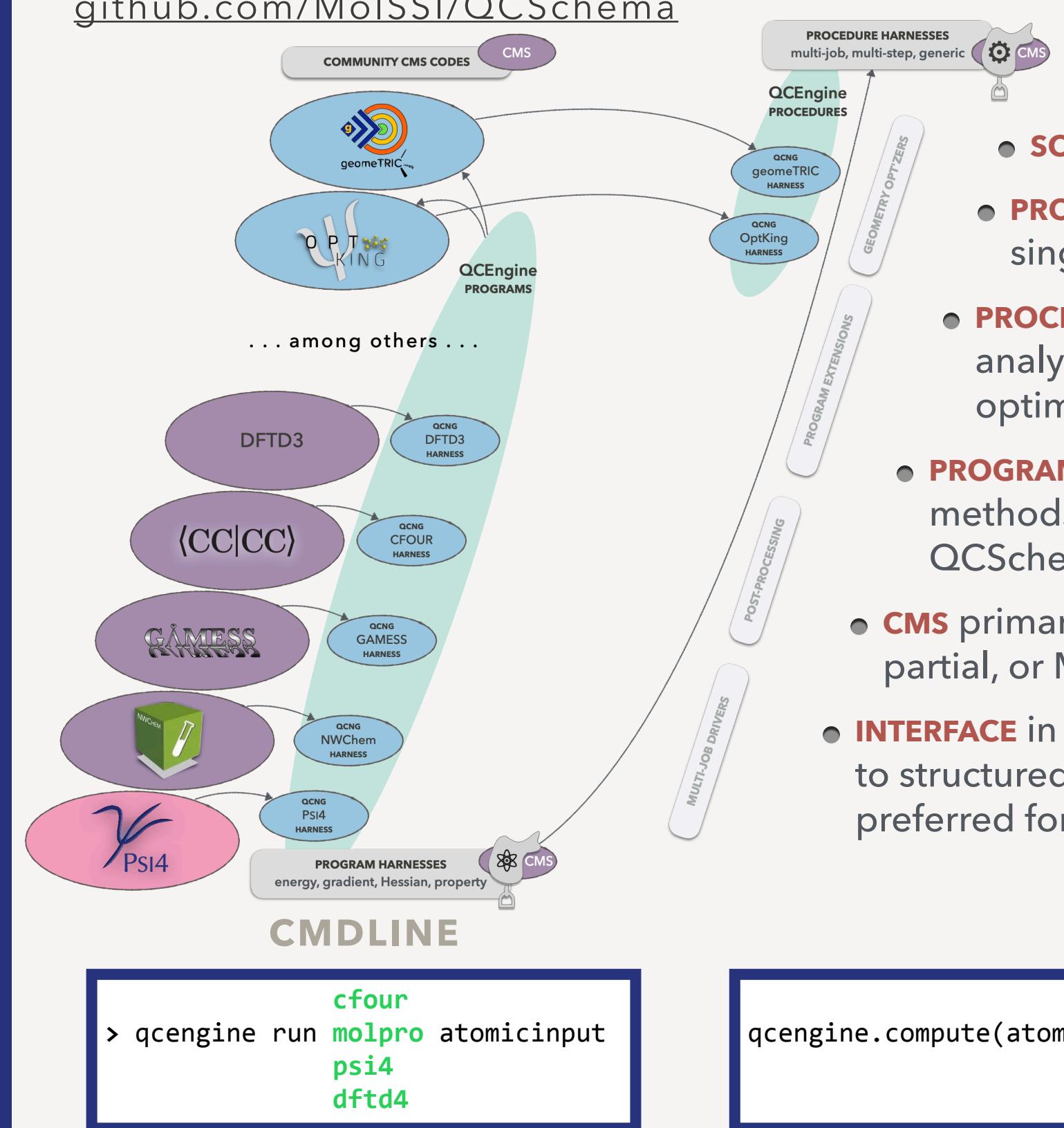
PRIMARY SCHEMAS DEFINED

- MOLECULE** Cartesian, atomic units (AU) based specification.
- ATOMICINPUT** job directive for analytic single-point – energy, gradient, Hessian, or property – as defined by **DRIVER**.
- ATOMICRESULT** append to **ATOMICINPUT** with output, simple scalar and array results, and wavefunction results.
- PROVENANCE** who wrote the data – program, version info.
- BASISSET** ~BSE. allows **WAVEFUNCTION** data in CCA ordering.
- OPTIMIZATIONINPUT/RESULT** job directive for abstract generic optimizer. calls generic QC program for gradient compute.



QCENGINE

github.com/MolSSI/QCSCHEMA



- SCHEMA** runner
- PROGRAMS** any analytic single point from CMS code
- PROCEDURES** anything except analytic single point. Only optimizers so far.
- PROGRAM CHECKS** indicate some methods accessible through QCSCHEMA
- CMS** primarily QM but also SE, MM, partial, or ML
- INTERFACE** in variety of ways from API to structured data to regex. Former preferred for numerical precision.

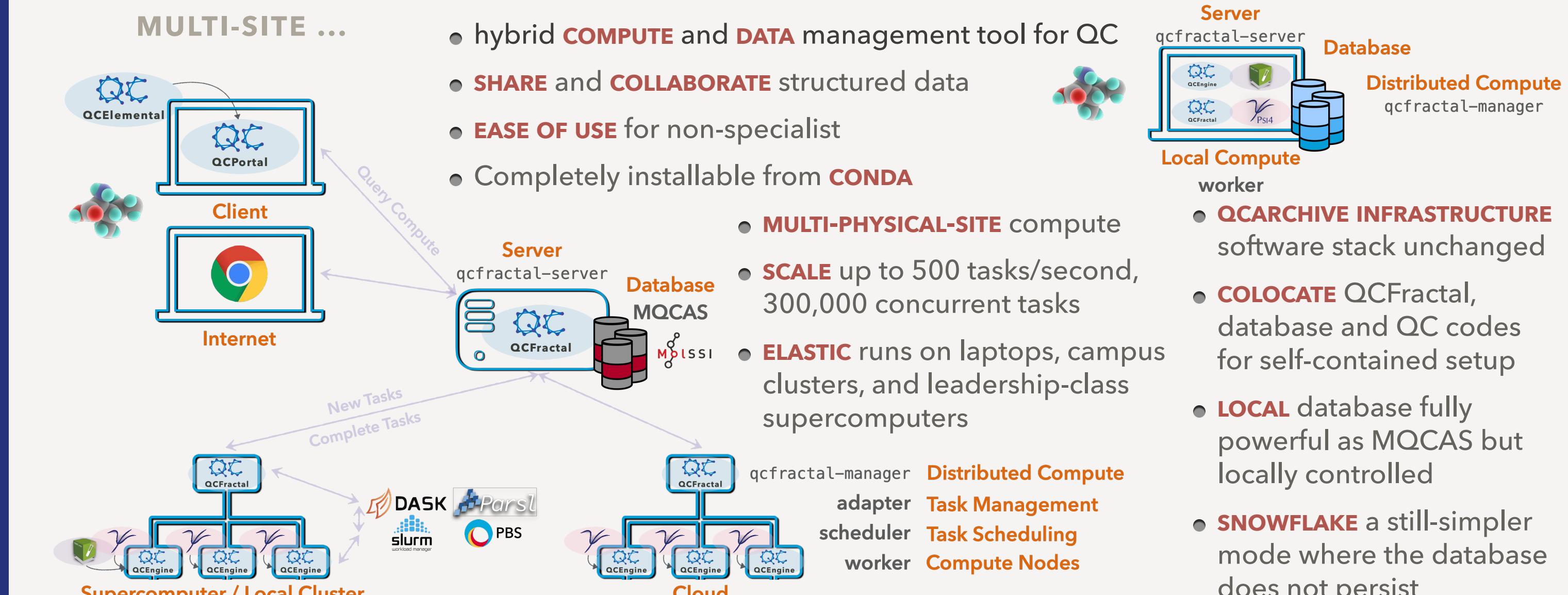
PYAPI

```
> qcengine run molpro atomicinput
psi4
dftd4
```

QCFRACAL

github.com/MolSSI/QCFractal

MULTI-SITE ...



... OR SINGLE-BOX

- Server** qcfractal-server
- Database** Database
- Distributed Compute** qcfractal-manager
- Local Compute** worker
- QCARCHIVE INFRASTRUCTURE** software stack unchanged
- COLOCATE** QCFractal, database and QC codes for self-contained setup
- LOCAL** database fully powerful as MQCAS but locally controlled
- SNOWFLAKE** a still-simpler mode where the database does not persist