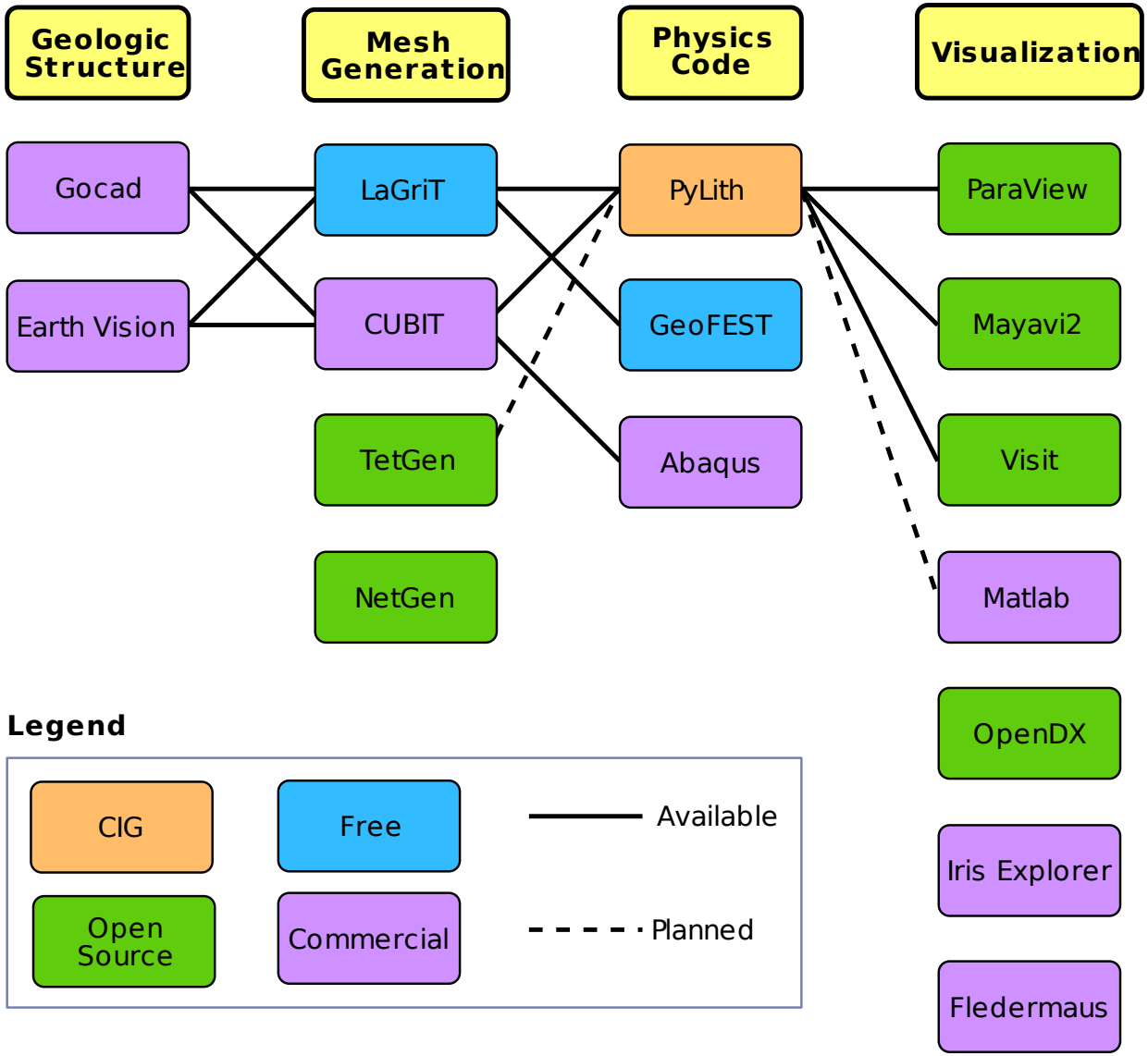
Crustal Deformation Modeling

Overview of workflow for typical research problem



Computational Infrastructure for Geodynamics

Ingredients for Running PyLith

- Simulation parameters
- Finite-element mesh
  - Mesh exported from LaGriT
  - Mesh exported from CUBIT
  - Mesh constructed by hand (PyLith mesh ASCII format)
- Spatial databases for physical properties, boundary conditions, and rupture parameters
  - SCEC CVM-H or USGS Bay Area Velocity model
  - Simple ASCII files

Spatial Databases

User-specified field/value in space

- Examples
  - Uniform value for Dirichlet (0-D)
  - Piecewise linear variation in tractions for Neumann BC (1-D)
  - SCEC CVM-H seismic velocity model (3-D)
- Generally independent of discretization for problem
- Available spatial databases

UniformDB Optimized for uniform value
SimpleDB Simple ASCII files (0-D, 1-D, 2-D, or 3-D)
SCECCVMH SCEC CVM-H seismic velocity model v5.3
ZeroDispDB Special case of UniformDB

Features in PyLith 1.4

Enhancements and new features in blue

- Time integration schemes
  - Implicit time stepping for quasi-static problems
  - Explicit time stepping for dynamic problems
- Bulk constitutive models
  - Elastic model (1-D, 2-D, and 3-D)
  - Linear and Generalized Maxwell viscoelastic models (3-D)
  - Power-law viscoelastic model (3-D)
- Boundary and interface conditions
  - Time-dependent Dirichlet boundary conditions
  - Time-dependent Neumann (traction) boundary conditions
  - Absorbing boundary conditions
  - Kinematic (prescribed slip) fault interfaces w/multiple ruptures
  - Time-dependent point forces
  - Gravitational body forces

Features in PyLith 1.4 (cont.)

Enhancements and new features in blue

- Automatic and user-controlled time stepping
- Ability to specify initial stress state
- Importing meshes
  - LaGriT: GMV/Pset
  - CUBIT: Exodus II
  - ASCII: PyLith mesh ASCII format (intended for toy problems only)
- Output: VTK files
  - Solution over volume
  - Solution over surface boundary
  - State variables (e.g., stress and strain) for each material
  - Fault information (e.g., slip and tractions)
- Automatic conversion of units for all parameters

PyLith 1.4: Under-the-hood Improvements

- General cleanup of C++ code
- Pyrex/pyrexembed replaced by SWIG
  - Greatly simplifies creating Python bindings for C++ objects
  - SWIG generated files included in source distribution
  - User-defined spatial databases and bulk constitutive models
- Automatic nondimensionalization of problem
  - User supplies pressure, time, and length scale of problem
  - All parameters nondimensionalized appropriately
  - Eliminates need to condition terms in sparse matrix
  - Restores symmetry of sparse matrix (reduces memory use)
- Integration with PETSc Scalable Nonlinear Equations Solvers
  - Disp. increment formulation for implicit and dynamic time-stepping

Time-Dependent Boundary Conditions

Dirichlet, Neumann, and Point Forces

$$f(\vec{x}) =$$

$f_0(\vec{x})+$	`db_initial`
$\dot{f}_1(\vec{x})(t - t_1(\vec{x}))+$	`db_rate`
$f_2(\vec{x})a(t - t_2(\vec{x}))$	`db_change`

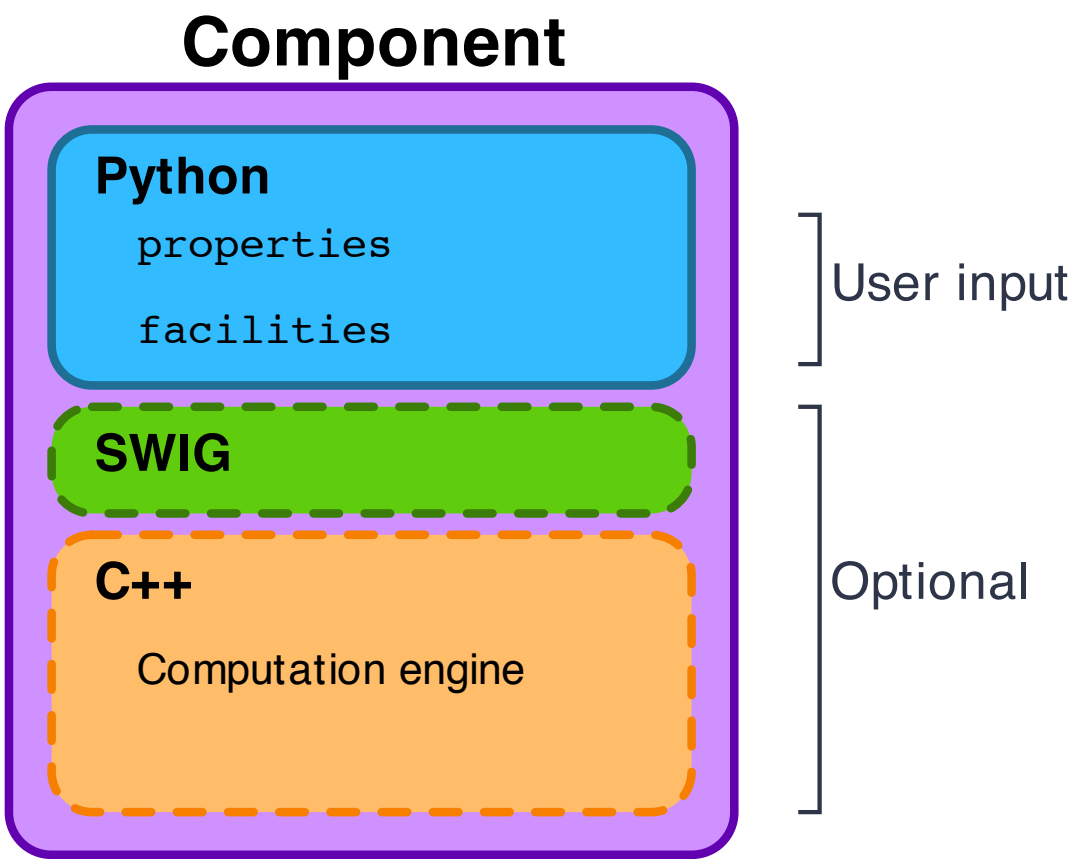
db_initial Initial value (constant in time)

db_rate Constant rate of change (spatially variable start time)

db_change Time history (spatially variable amplitude and start time)

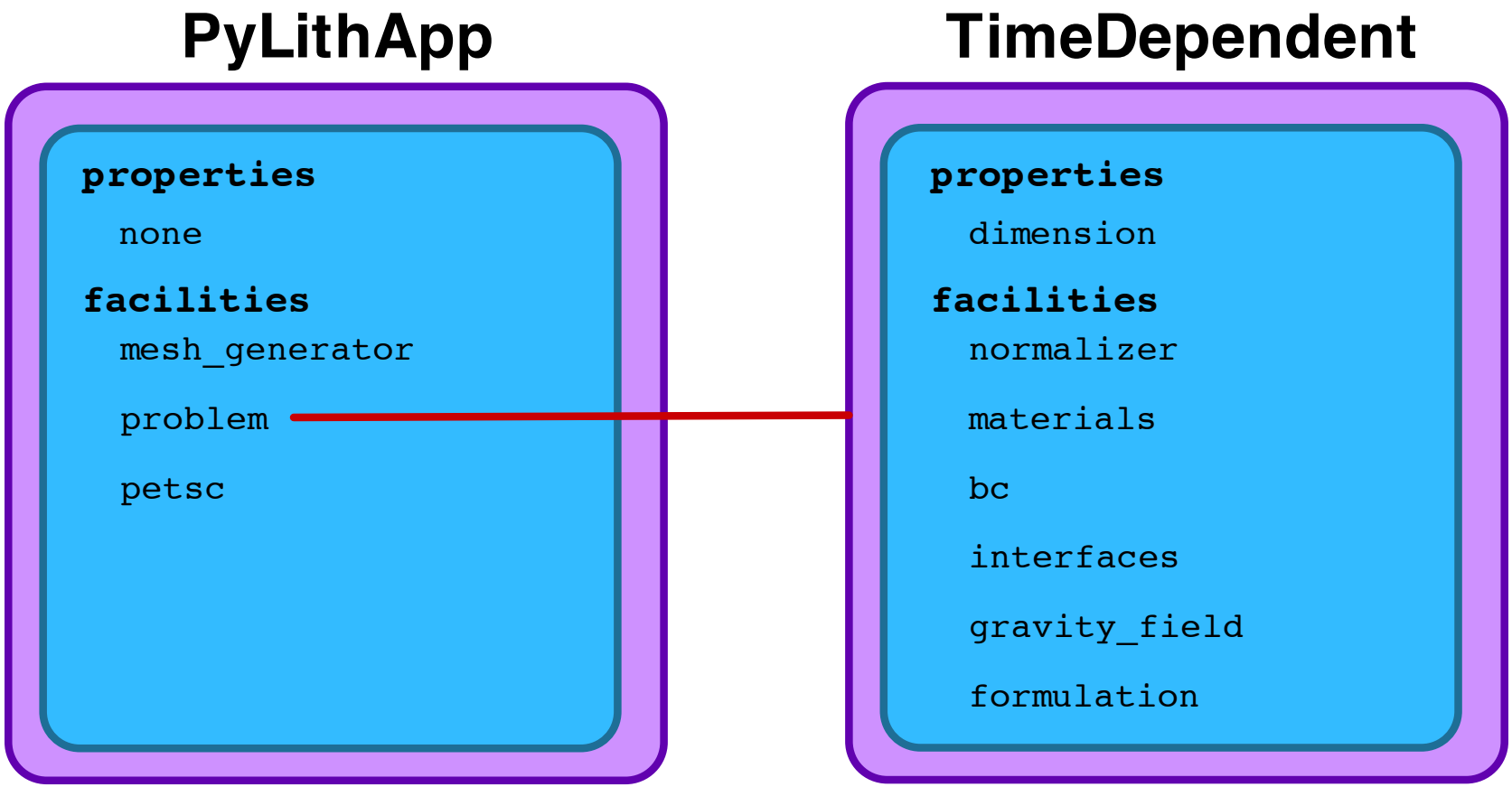
PyLith as a Hierarchy of Components

Components are the basic building blocks



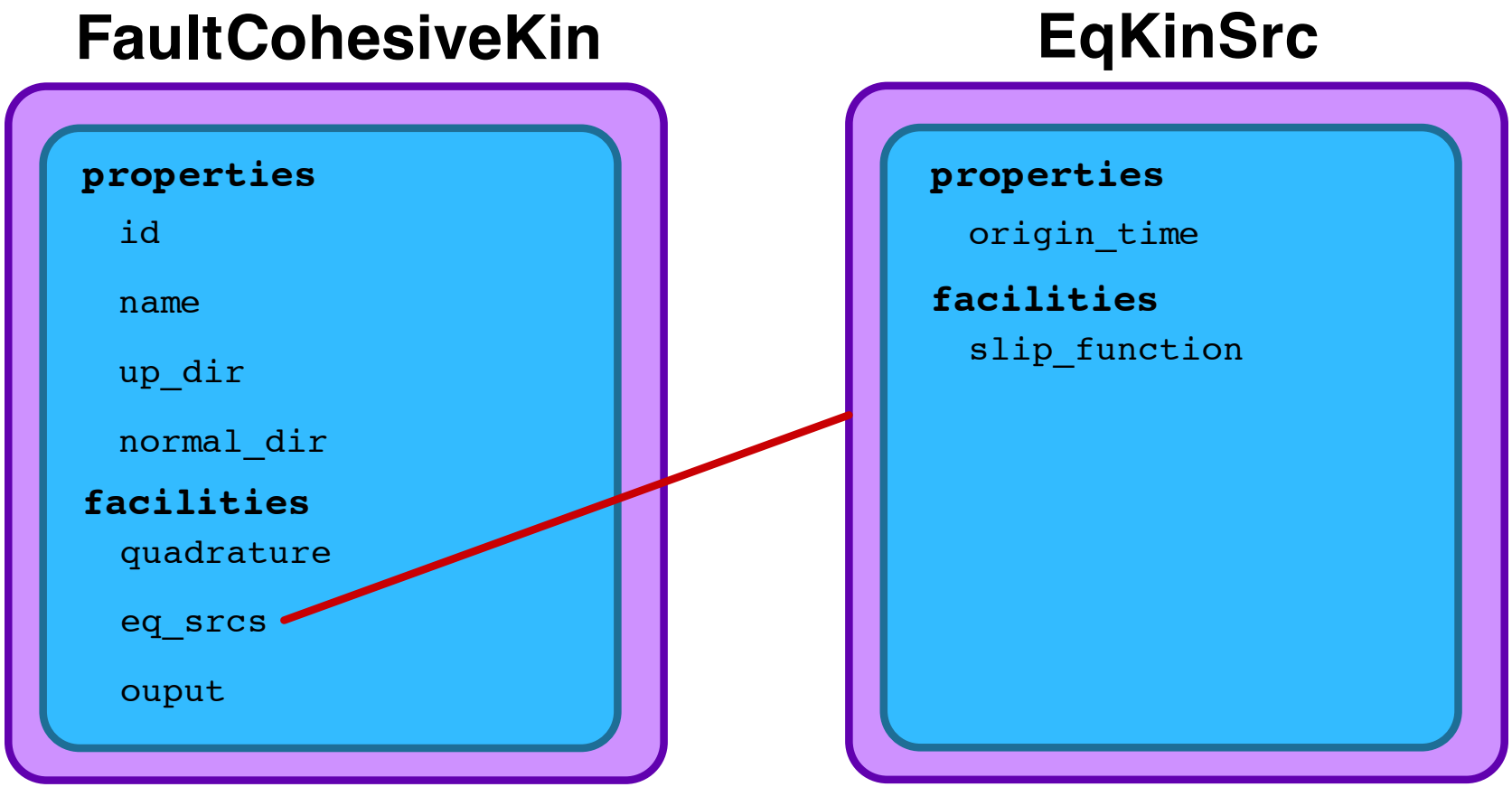
PyLith as a Hierarchy of Components

PyLith Application and Time-Dependent Problem



PyLith as a Hierarchy of Components

Fault with kinematic (prescribed slip) earthquake rupture



PyLith Application Flow

PyLithApp

```
main()
  mesher.create()
  problem.initialize()
  problem.run()
```

TimeDependent (Problem)

```
initialize()
  formulation.initialize()

run()
  while (t < totalTime)
    dt = formulation.getTimeStep()
    formulation.prestep()
    formulation.step()
    formulation.poststep()
```

Implicit (Formulation)

```
initialize()

prestep()
  set constraints

step()
  calculate residual
  solve for displacement increment

poststep()
  update displacement field
  write output
```

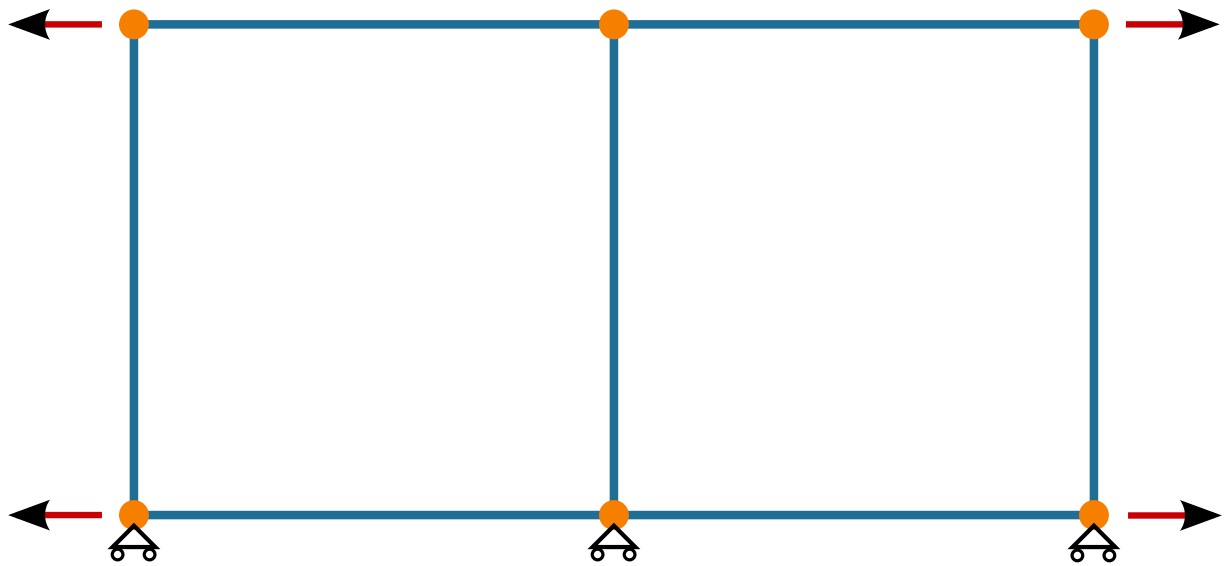
Computational Infrastructure for Geodynamics

Ingredients for Running PyLith

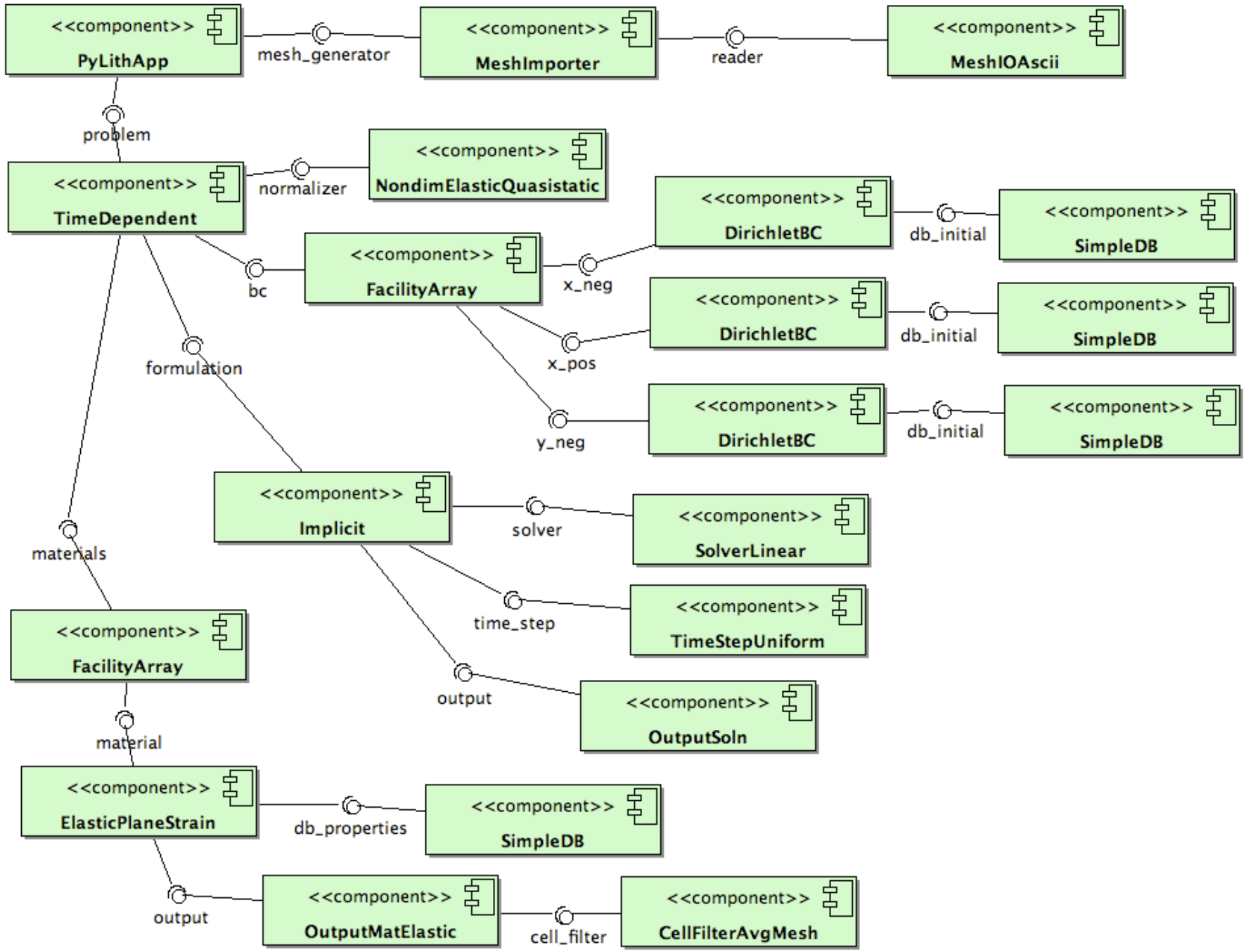
- Simulation parameters
  - `.cfg` ASCII files
  - `pylithapp.cfg` always read if it exists
  - Command line arguments
- Finite-element mesh
  - Mesh exported from LaGriT
  - Mesh exported from CUBIT
  - Mesh constructed by hand (PyLith mesh ASCII format)
- Spatial databases for physical properties, boundary conditions, and rupture parameters

Example: `twocells/twoquad4 axialdisp.cfg`

Axial extension via prescribed displacements



Example: `twocells/twoquad4 axialdisp.cfg`



Computational Infrastructure for Geodynamics

Example: `twocells/twoquad4 axialdisp.cfg`

Input

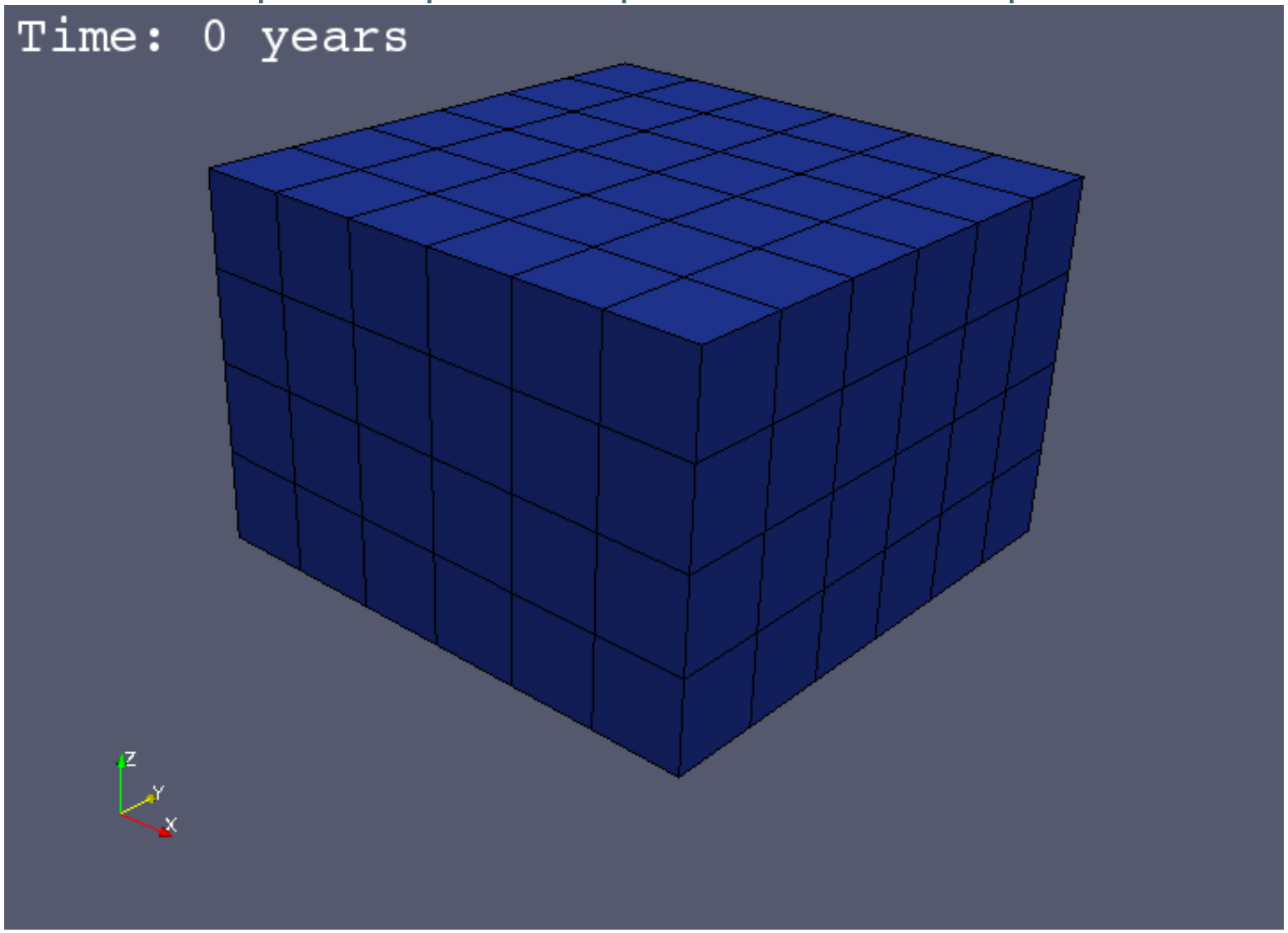
- Simulation parameters
  - `pylithapp.cfg`
  - `axialdisp.cfg`
- ASCII Mesh: `twoquad4.mesh`
- Spatial databases
  - `matprops.spatialdb`
  - `axialdisp.spatialdb`

Output

- Displacement field
  - `axialdisp_t000000.vtk`
- State variables
  - `axialdisp-statevars_info.vtk` (physical properties)
  - `axialdisp-statevars_t000000.vtk` (stress and strain)

Example: `3d/hex8 savageprescott.cfg`

Creep and repeated rupture on a strike-slip fault



Example: `3d/hex8 savageprescott.cfg`

Input

- Simulation parameters
  - `pylithapp.cfg`
  - `savageprescott.cfg`
- Mesh: `box_hex8_1000m.exo`
- Spatial databases
  - `mat_elastic.spatialdb`
  - `mat_maxwell.spatialdb`
  - `finalslip_rupture.spatialdb`
  - `sliptime.spatialdb`
  - `sliprate_creep.spatialdb`

Output

- Displacement field
  - `savageprescott_tNNNN.vtk`
  - `savageprescott-groundsurf_tNNNN.vtk`
- State variables
  - `savageprescott-elastic_info.vtk`
  - `savageprescott-elastic_tNNNN.vtk`
  - `savageprescott-viscoelastic_info.vtk`
  - `savageprescott-viscoelastic_tNNNN.vtk`
- Fault
  - `savageprescott-fault_info.vtk`
  - `savageprescott-fault_tNNNN.vtk`

Useful Tips/Tricks

- `pylithinfo [--verbose] [PyLith args]`
  Dumps all parameters with their current values to text file

- Command line arguments
  - `--help`
  - `--help-components`
  - `--help-properties`
  - `--petsc.start_in_debugger` (run in xterm)
  - `--nodes=N` (to run on N processors on local machine)

- PyLith User Manual

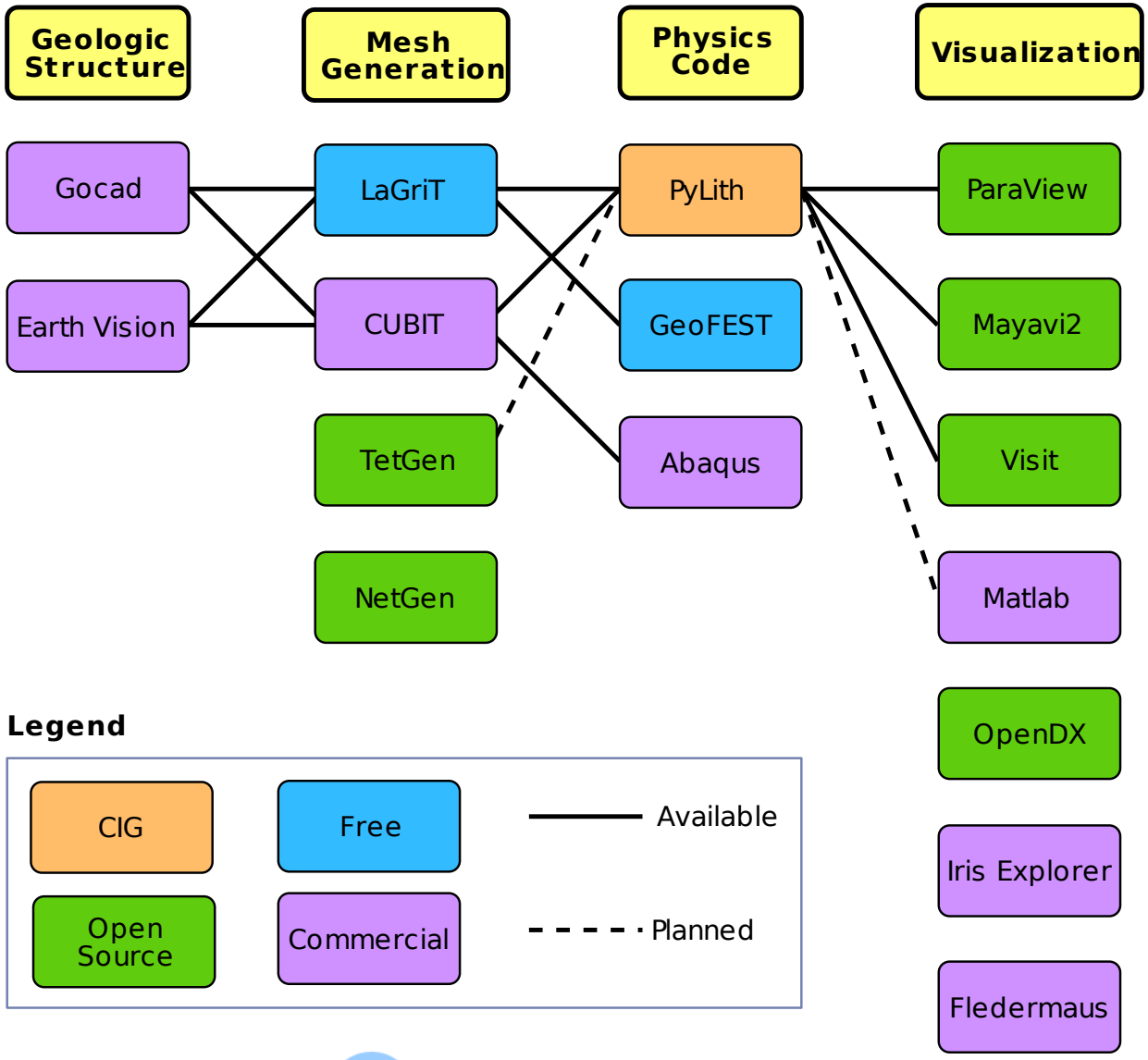
- CIG Short-Term Tectonics mailing list
  `cig-short@geodynamics.org`

- CIG bug tracking system
  `http://www.geodynamics.org/roundup`

Computational Infrastructure for Geodynamics

Crustal Deformation Modeling

Overview of workflow for typical research problem



Computational Infrastructure for Geodynamics