

# Extending PyLith

Brad Aagaard and Charles Williams  
Matthew Knepley and Surendra Somala



June 15, 2010

# How can PyLith be extended?

- Any component can be replaced
  - Custom spatial databases
  - Bulk constitutive models
  - Fault constitutive models
  - Solvers
  - Boundary conditions
- Difficult to change: governing equations
  - Coupled fluid flow
  - Coupled heat flow

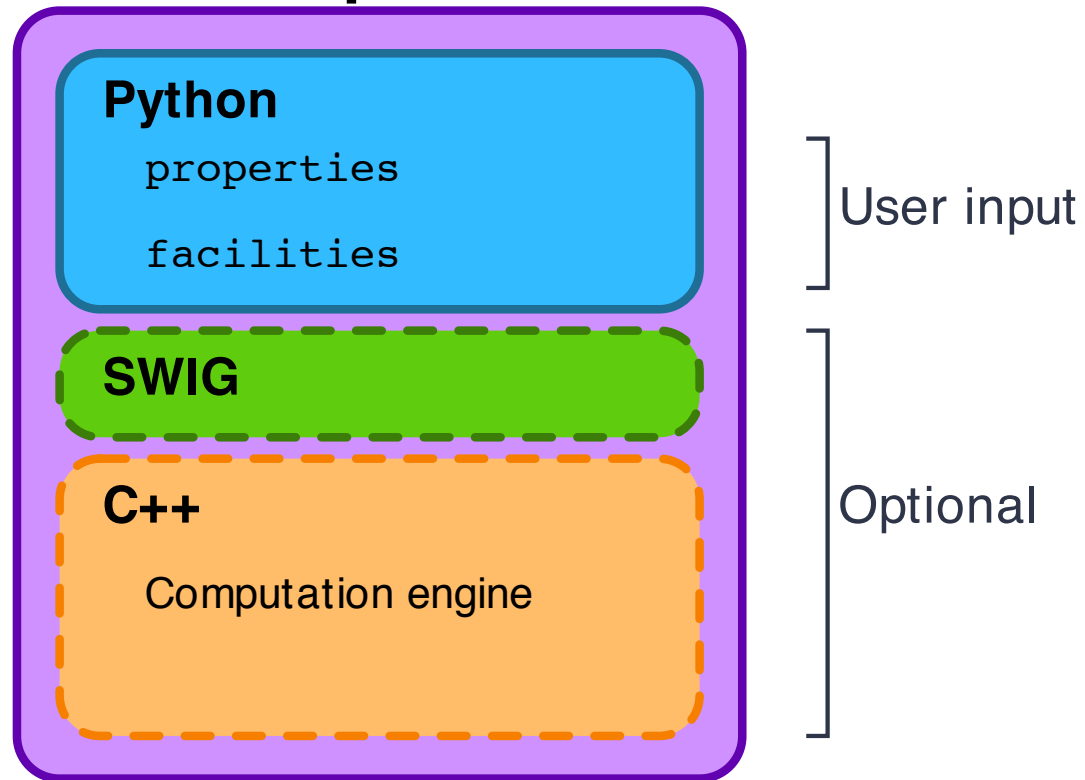
# Prerequisites for Adding Components

Recommended for advanced users only

- Comfortable working with
  - Makefiles
  - Python
  - C/C++
  - Object-oriented programming
- Good at pattern matching
- Desire to break new ground

# Anatomy of a PyLith Component

## Component



# Implementation of a PyLith Component

- Python object
  - Gather user input
  - Do high-level validation of parameters
  - Send user input to C++
- C++ object
  - Accept user input from Python
  - Implement interface required for component

# Example: Bulk Constitutive Model

Elastic plane strain w/state variables

- Python object
  - User defined parameters: none
  - Inherited from Elastic material
    - `db_properties`
    - `db_initial_state`
- C++ object
  - Define physical properties and state variables
  - Convert db parameters to values used internally
  - Nondimensionalization of parameters
  - Implement rheology
    - Calculate density, stresses, and elastic constants
- Verify functionality using unit tests