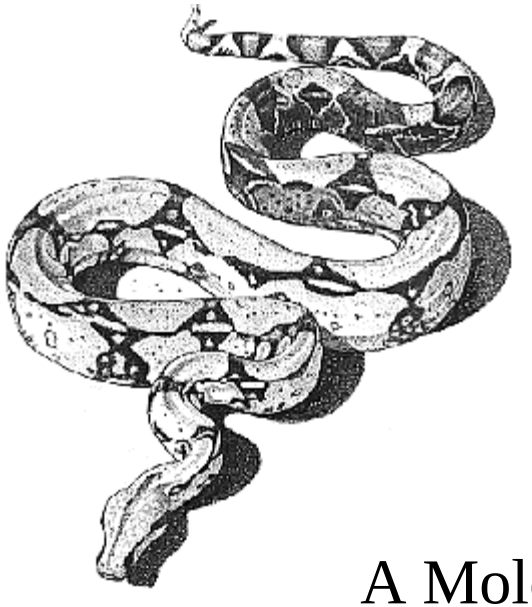


PD

A Molecular Mechanics Engine for Python

Mike Tyka & Jon Rea

Prof. Clarke & Dr. Richard Sessions
Dept. Biochemistry, University of Bristol

Concepts & Motivation

- Flexibility for the User *(do lots, code nothing)*
- Flexibility for Developer *(do lots, code little)*
- Focus on Science *(avoid unnecessary code)*
- Fast Execution *(avoid waiting for results)*

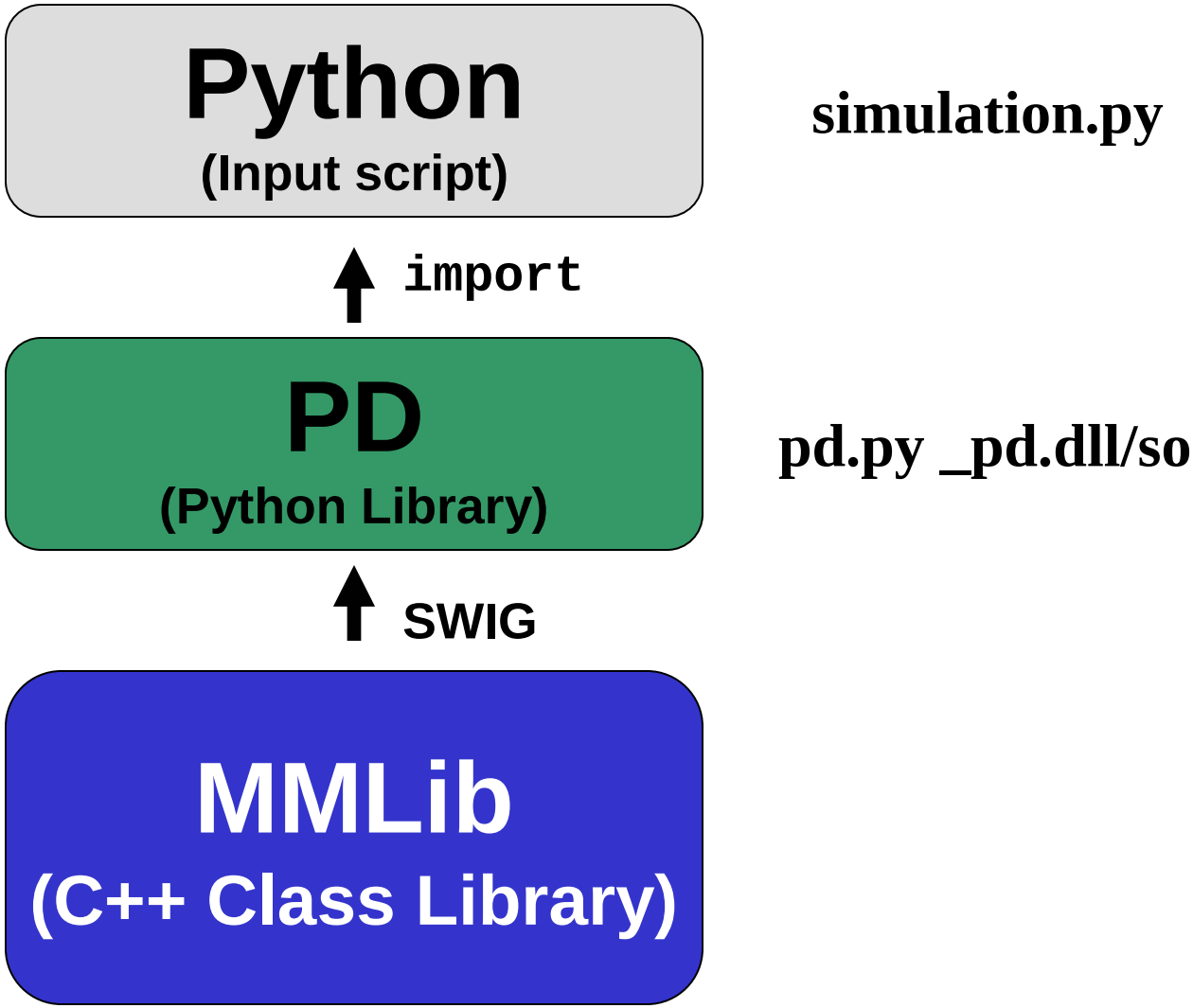
Concept & Motivation

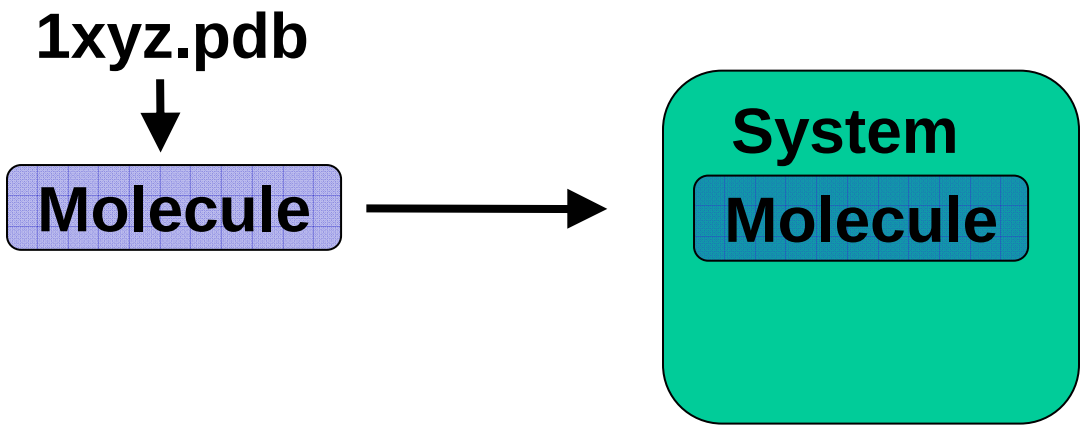
- Flexibility for the User
  - Simulation machinery exported into **Python**
  - Input script: minimal, simple, advanced or very complex
  - Syntax intuitive
  - Classes interact at runtime to give new algorithms
  - Python is widely used & talks to existing systems
  - Platform independence
- Flexibility for Developer
  - Object Oriented Design → Code re-usage
  - Inheritance for extending algorithms

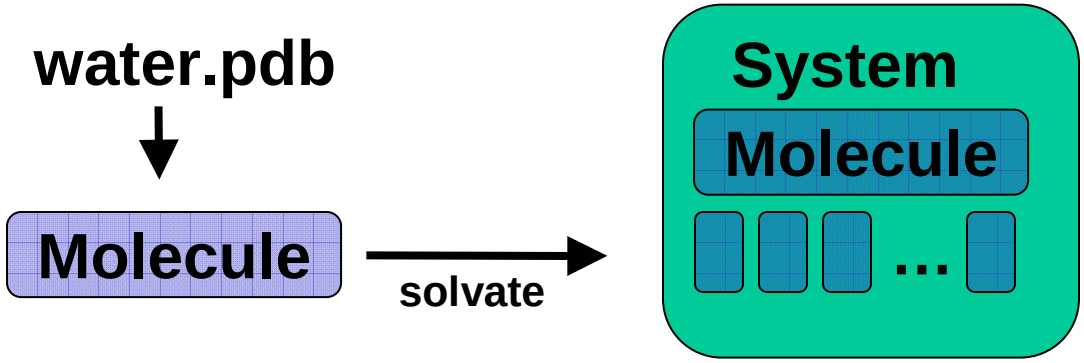
Concept & Motivation

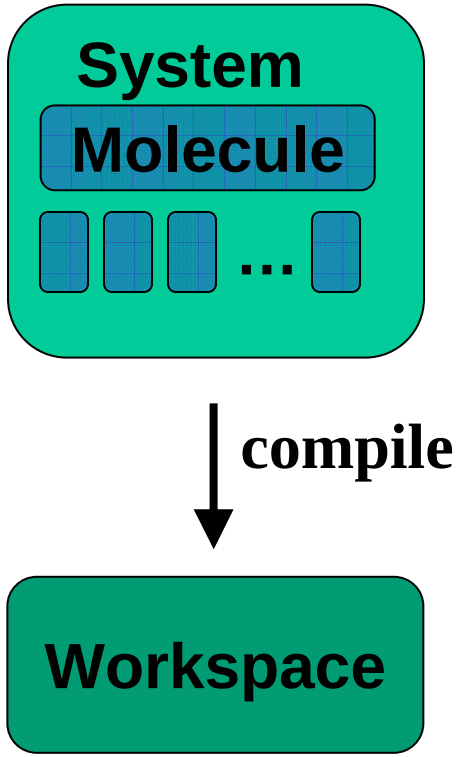
- Rapid Development
  - Interface generated *automatically* using SWIG
  - Template/Example classes for major types (e.g. Forcefields, Protocols)
  - Runtime linkage allows external modules
  - Perl, TCL, PHP, C# / ASP.NET…
- Fast Execution
  - **C++** - mature, widely used, powerful compilers
  - MPI in development

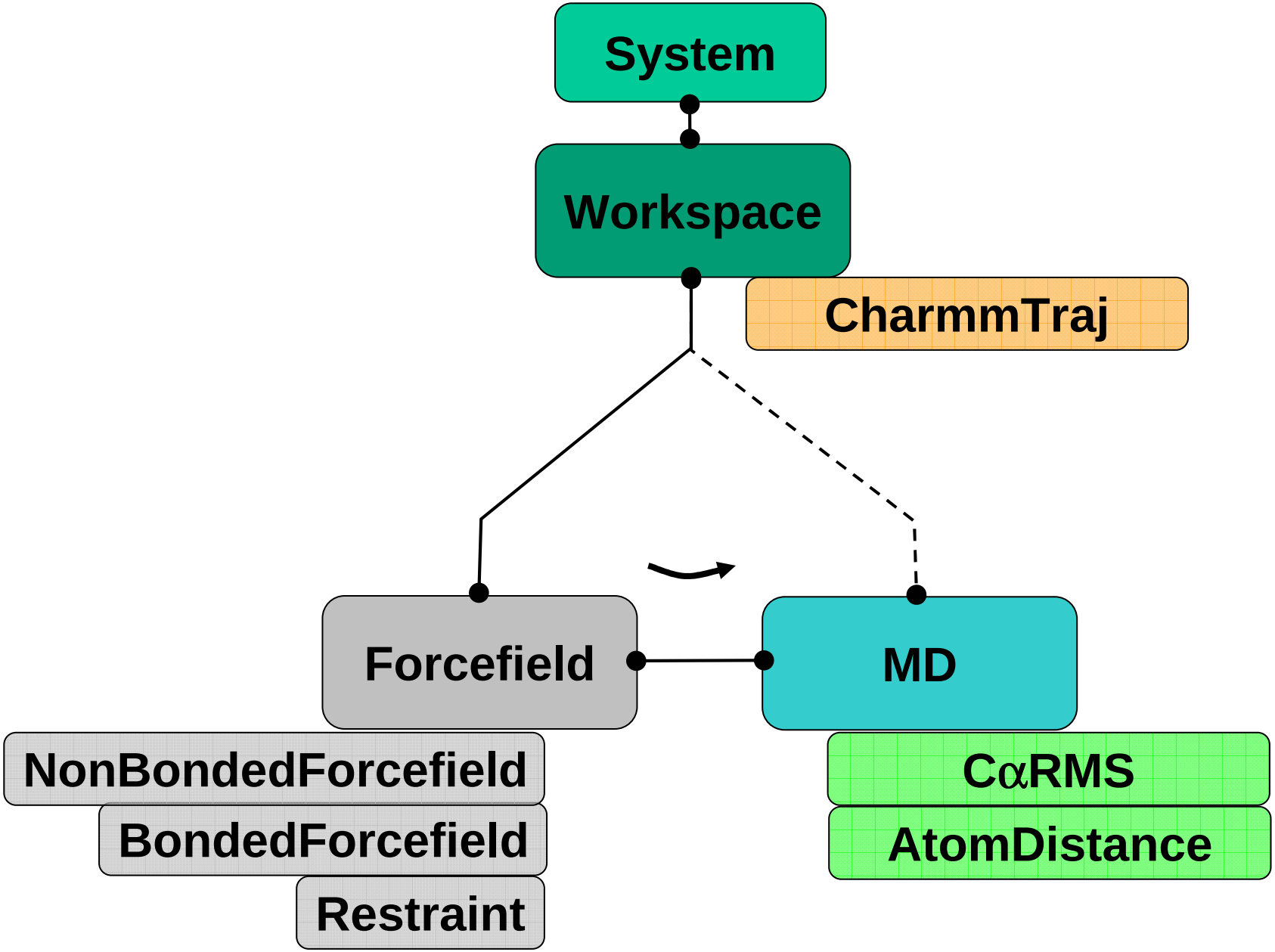
Basic PD Structure




1xyz.pdb
Molecule
System
Molecule


water.pdb
Molecule
solvate
System
Molecule
...


System
Molecule
...
compile
Workspace


System
Workspace
CharmmTraj
Forcefield
MD
NonBondedForcefield
BondedForcefield
Restraint
CαRMS
AtomDistance

Design Features

Flexibility for users

Advantages of Pythonisation

```
## Run 4 MD runs at increasing temperature

for T in [ 300, 350, 400, 500 ]:
  md.TargetTemp = Temp(T)
  md.run()
```

- **Python is a full blown object oriented language !!**
- **Python comes with "batteries included"**

Modularity

```
evaluator = Energy(ff)

moveset = MoveSet(wspace)
moveset.add( SidechainTorsionalMove())

mc = MonteCarlo(evaluator,moveset)
mc.Steps = 200
mc.TargetTemp = Temp(300)

mc.run()
```

→ **Classic Metropolis Montecarlo (Metropolis et. al, 1973)**

Modularity

```
evaluator = Energy(ff)

moveset = MoveSet(wspace)
moveset.add( SidechainTorsionalMove())

mc = MonteCarlo(evaluator,moveset)
mc.Steps = 200
mc.TargetTemp = TempLinear(300,10)

mc.run()
```

→ **Simulated Annealing**

Modularity

```
evaluator = Minimisation(ff)
evaluator.Steps = 200
moveset = MoveSet(wspace)
moveset.add( SidechainTorsionalMove())

mc = MonteCarlo(evaluator,moveset)
mc.Steps = 200
mc.TargetTemp = Temp(300)

mc.run()
```

→ **Montecarlo with Minimisation (Scheraga, 1973)**

Modularity

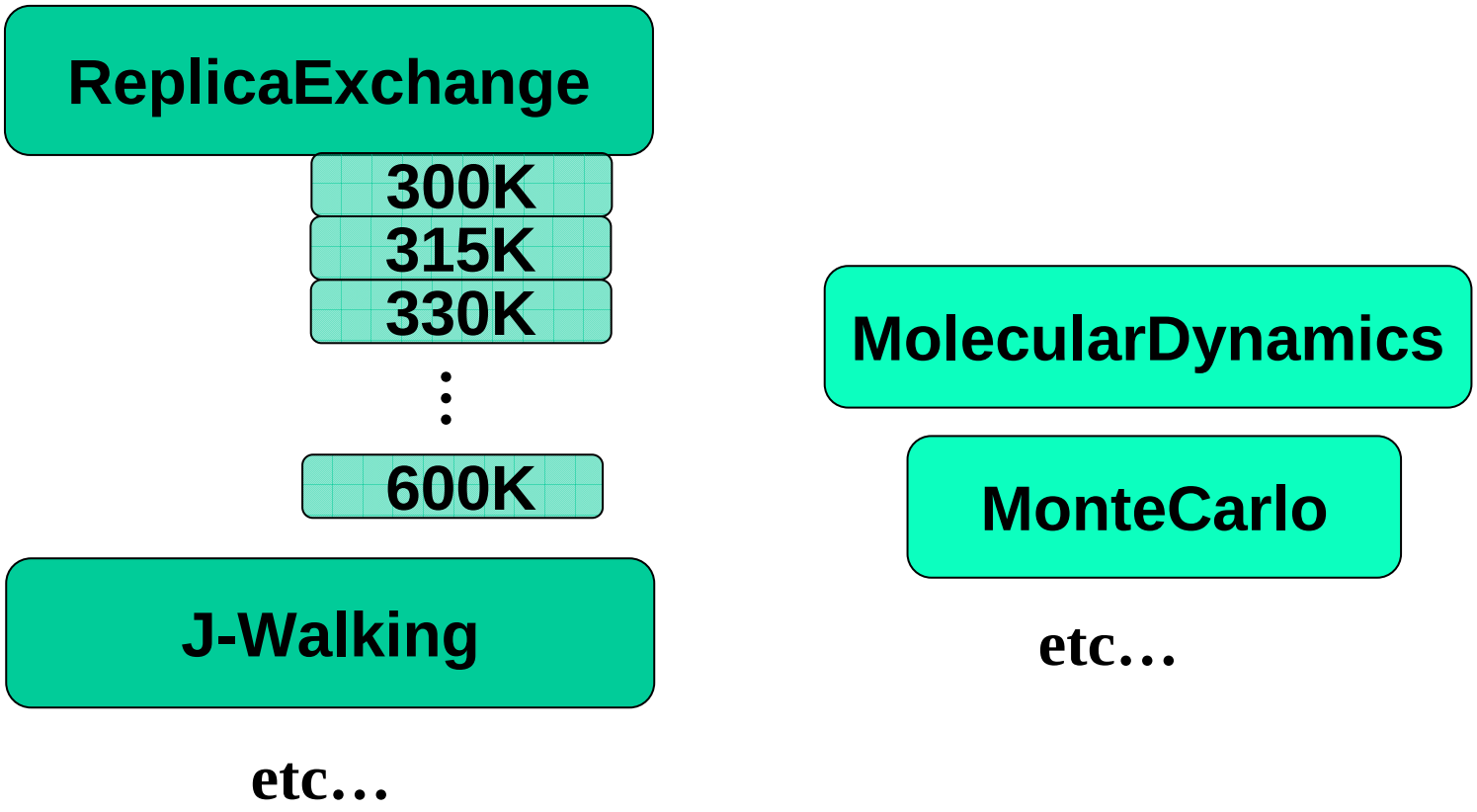
```
evaluator = MolecularDynamics(ff)
evaluator.Steps = 200
moveset = MoveSet(wspace)
moveset.add( SidechainTorsionalMove())

mc = MonteCarlo(evaluator,moveset)
mc.Steps = 200
mc.TargetTemp = Temp(300)

mc.run()
```

→ **Hybrid Monte Carlo (Duane et al., 1987)**

Design Features

Flexibility for developers

Extendibility I



e.g.: Replica Exchange Dynamics(Sugita et al., 1999)

Extendibility II

Object Oriented Design – Defines Major Classes and interations

Inheritance

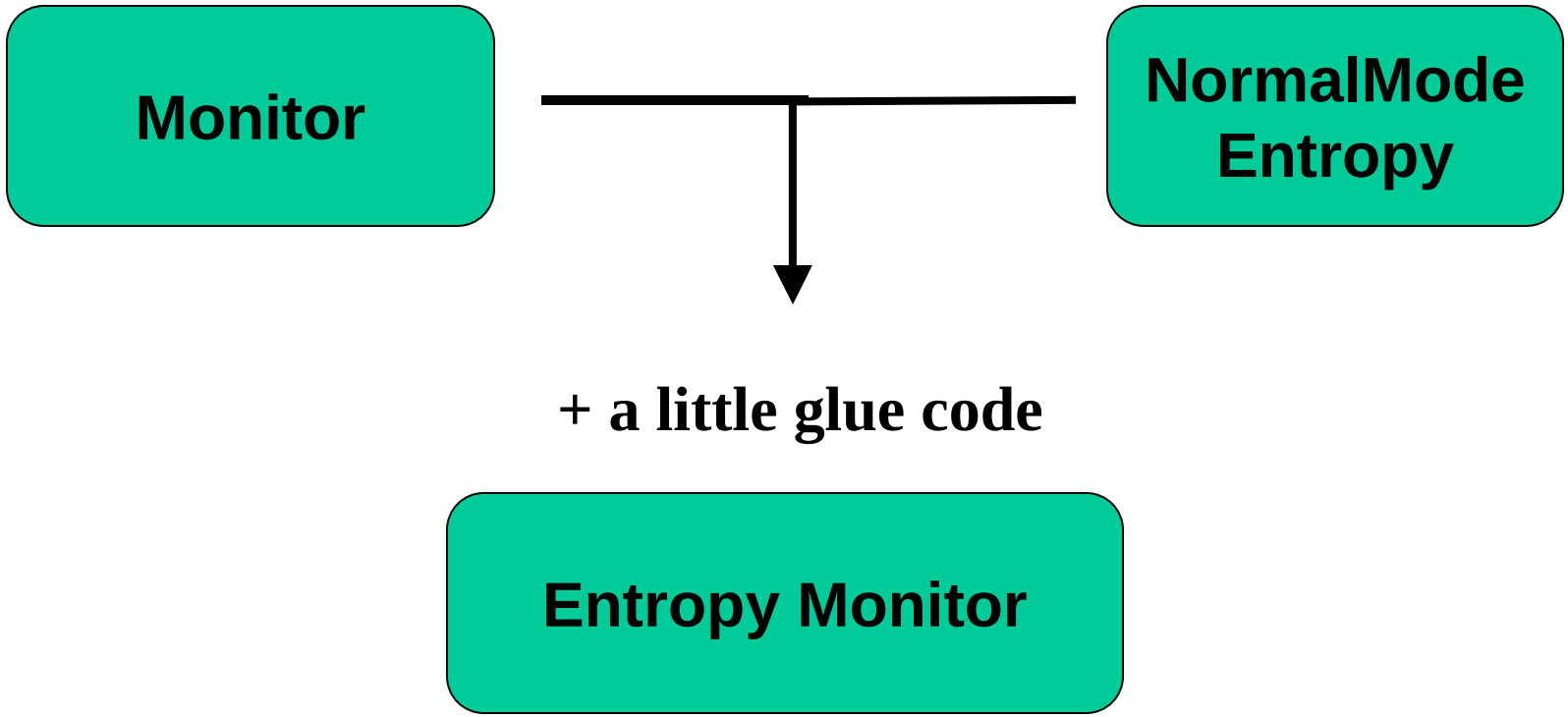
Only ~20 new lines of code !

SWIG automatically generates Python interface

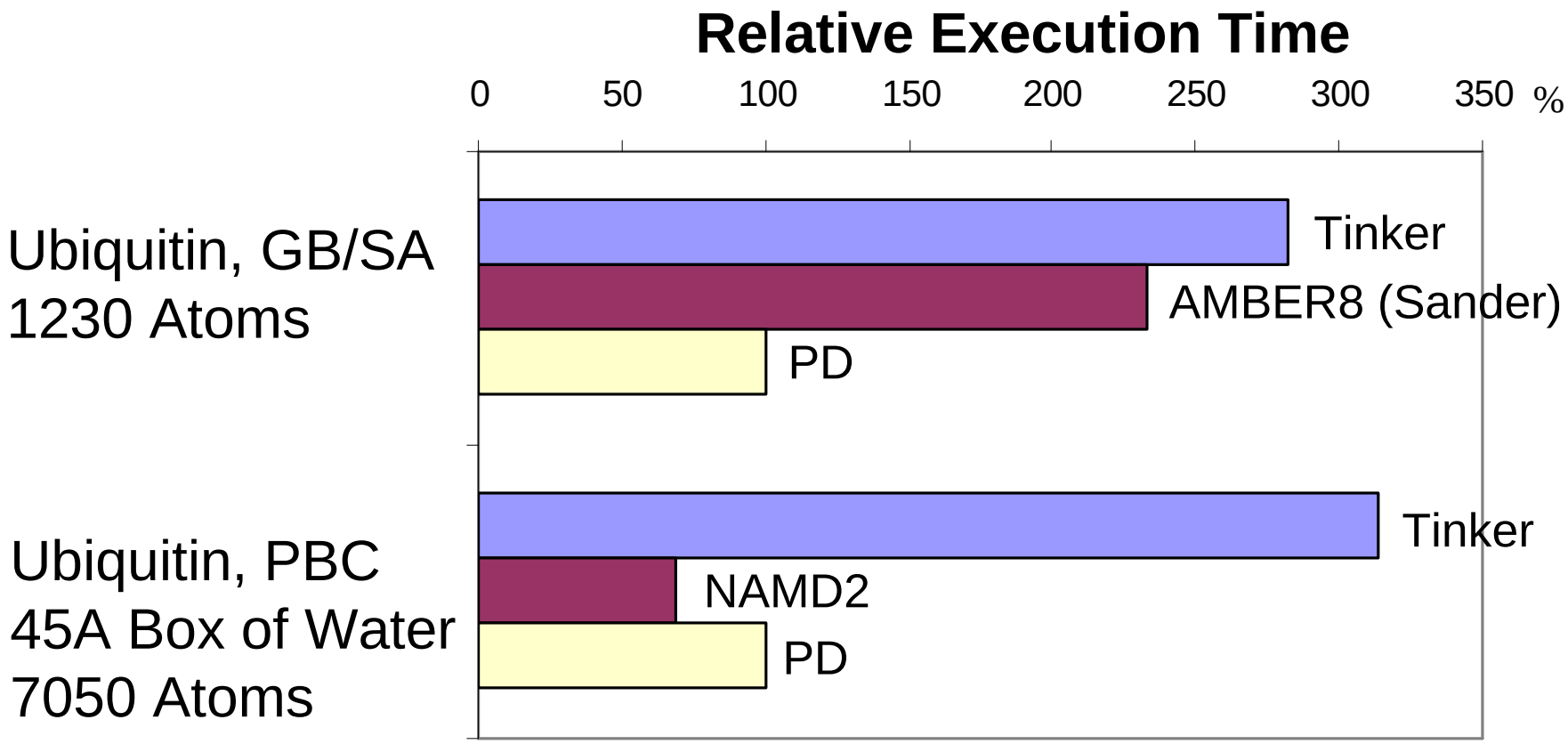
Extendibility III

Object Oriented Design – Defines Major Classes and interations

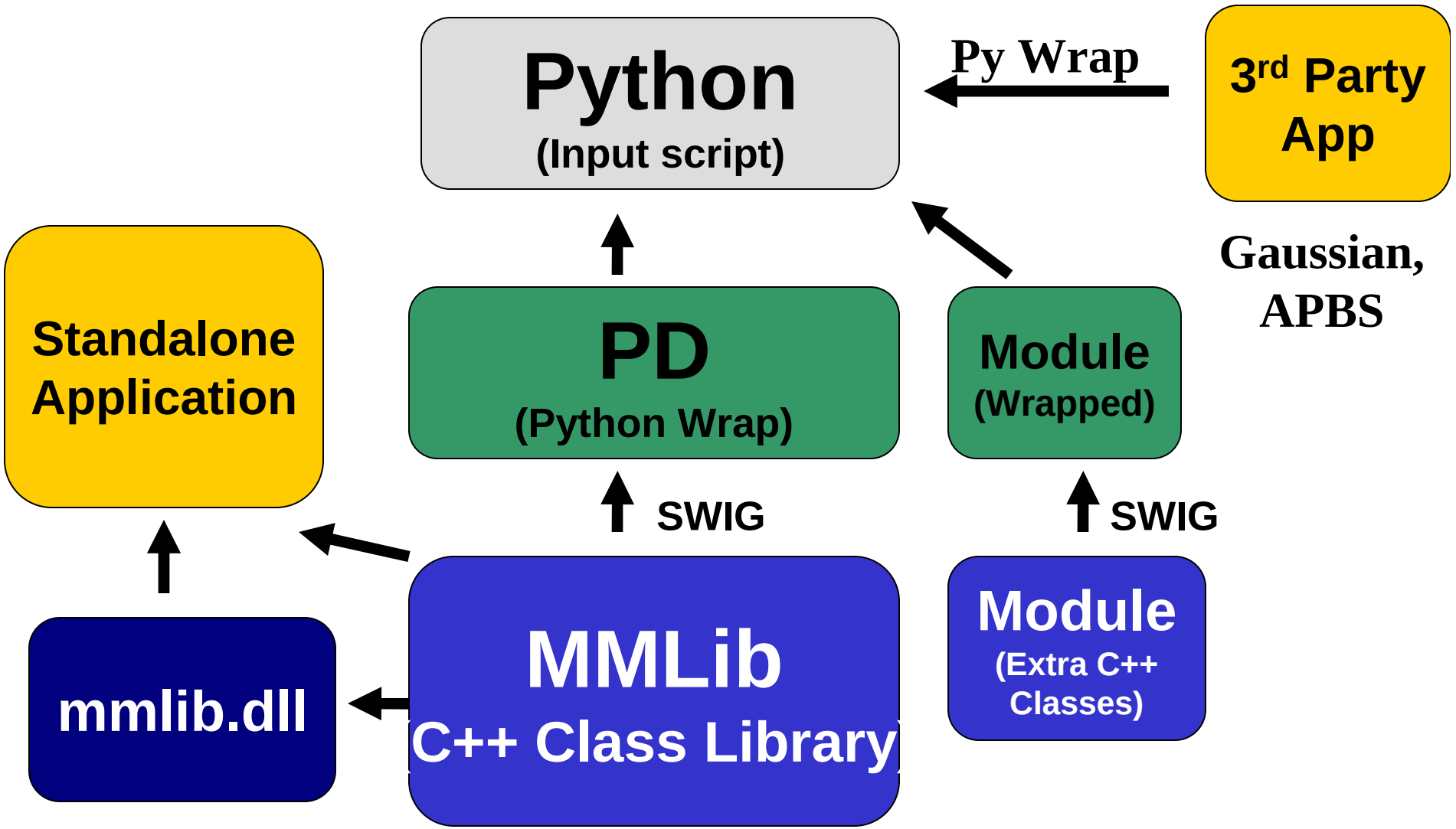
Multiple Inheritance



Preliminary Benchmarks



PD Structure - Advanced



Selected Features

- Implicit Solvation (GBSA/SASA), Explicit Solvation (Periodic Boundaries)
- AMBER, CHARMM, OPLS
- Molecular Dynamics, REMD, Monte Carlo
- Free Energy Methods (FEP,TI, Umbrella, Confinement Method)
- Homology Modelling, Loop Building
- Charmm DCD/PSF Output → Interacts with VMD & Chimera

Current Development

- Small Molecule Docking

- Multi-time stepping
- Particle Mesh Ewald
- MPI – Trivial parallelisation of REMD etc..
- MPI – Parallelisation of nonbonded and bonded interactions

- PyMol / Chimera Interactive Interface

Conclusion

Flexible Simulation and Development Environment

Planned Release Date: ~Feb/March 07

Licence: Academic free
Commercial $$

m.tyka@bris.ac.uk
www.bris.ac.uk/biochemistry/cpfg/pd.htm

Acknowledgements

Jon Rea

Jon Crisp
Tom Piggot

Tony Clarke
Richard Sessions

Dept. of Biochemistry

m.tyka@bris.ac.uk
www.bris.ac.uk/biochemistry/cpfg/pd.htm