Introduction to OpenSim's Architecture and API

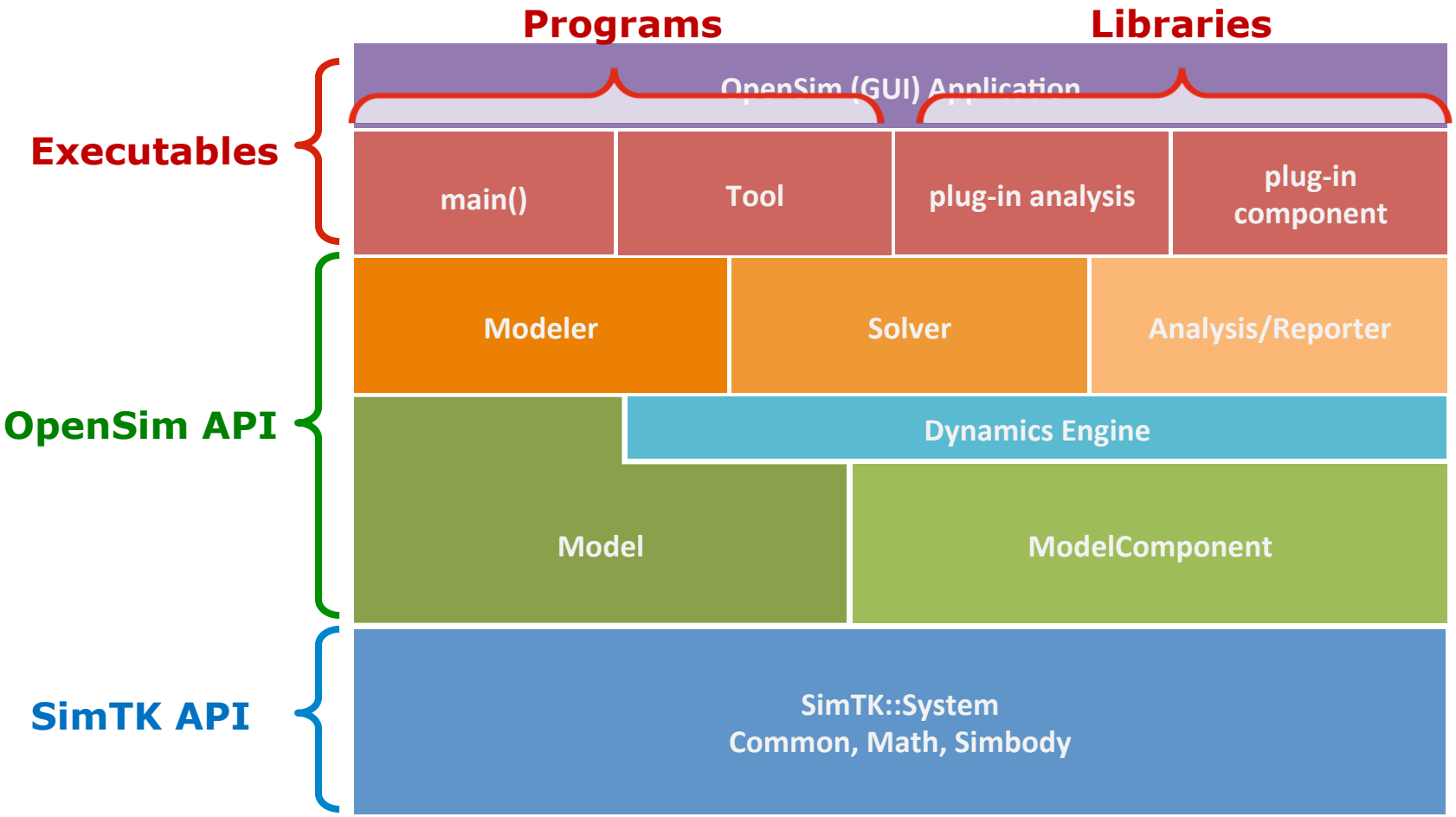
Goals of the OpenSim API:

Enable OpenSim users and developers to develop their own custom musculoskeletal models and simulations.

To do so one needs an understanding of:

1. How the OpenSim software is organized
2. What is the OpenSim Application Programming Interface (API)?

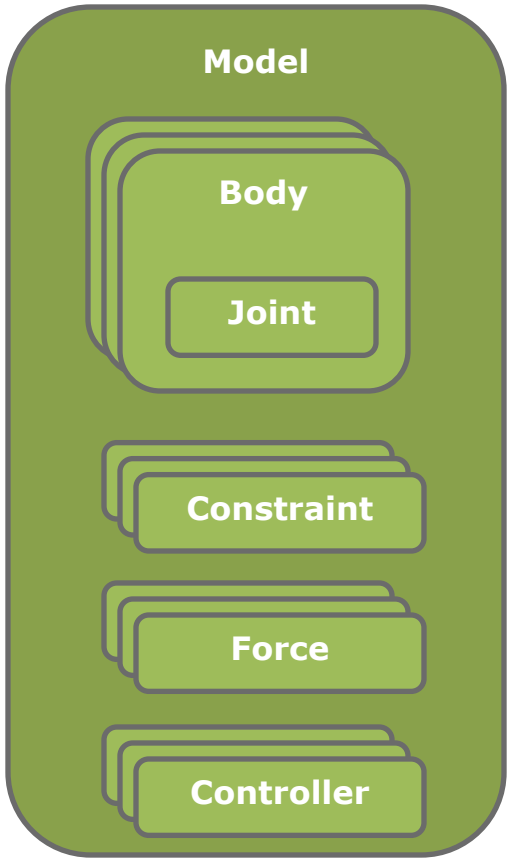
OpenSim Architecture: Interface Layers



OpenSim Model Class Structure:

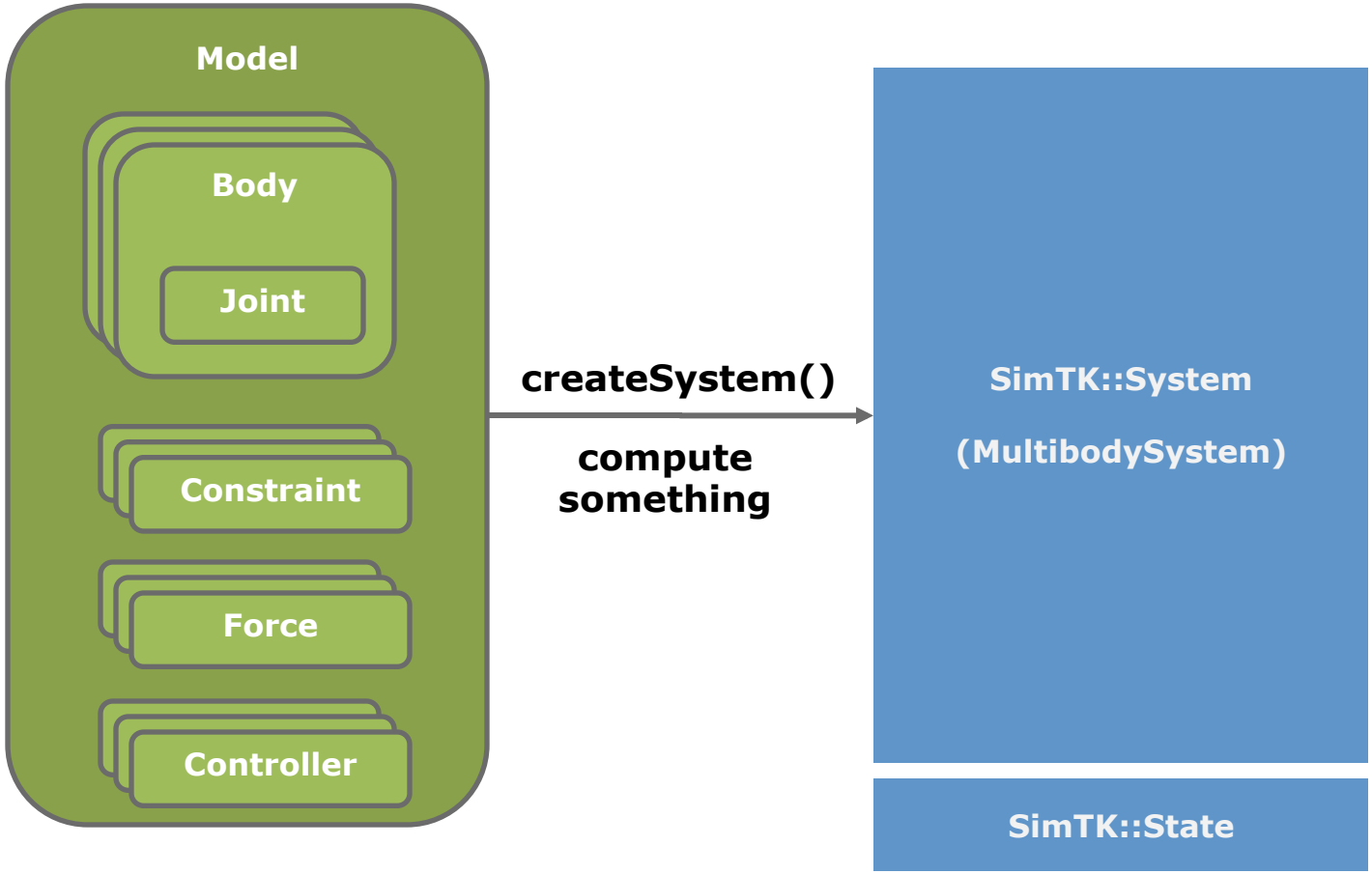
Model = physics of a biomechanical system comprised of physical elements: bodies, forces, constraints …

An OpenSim model is a bag of components corresponding to a physical system and maps to an underlying computational system



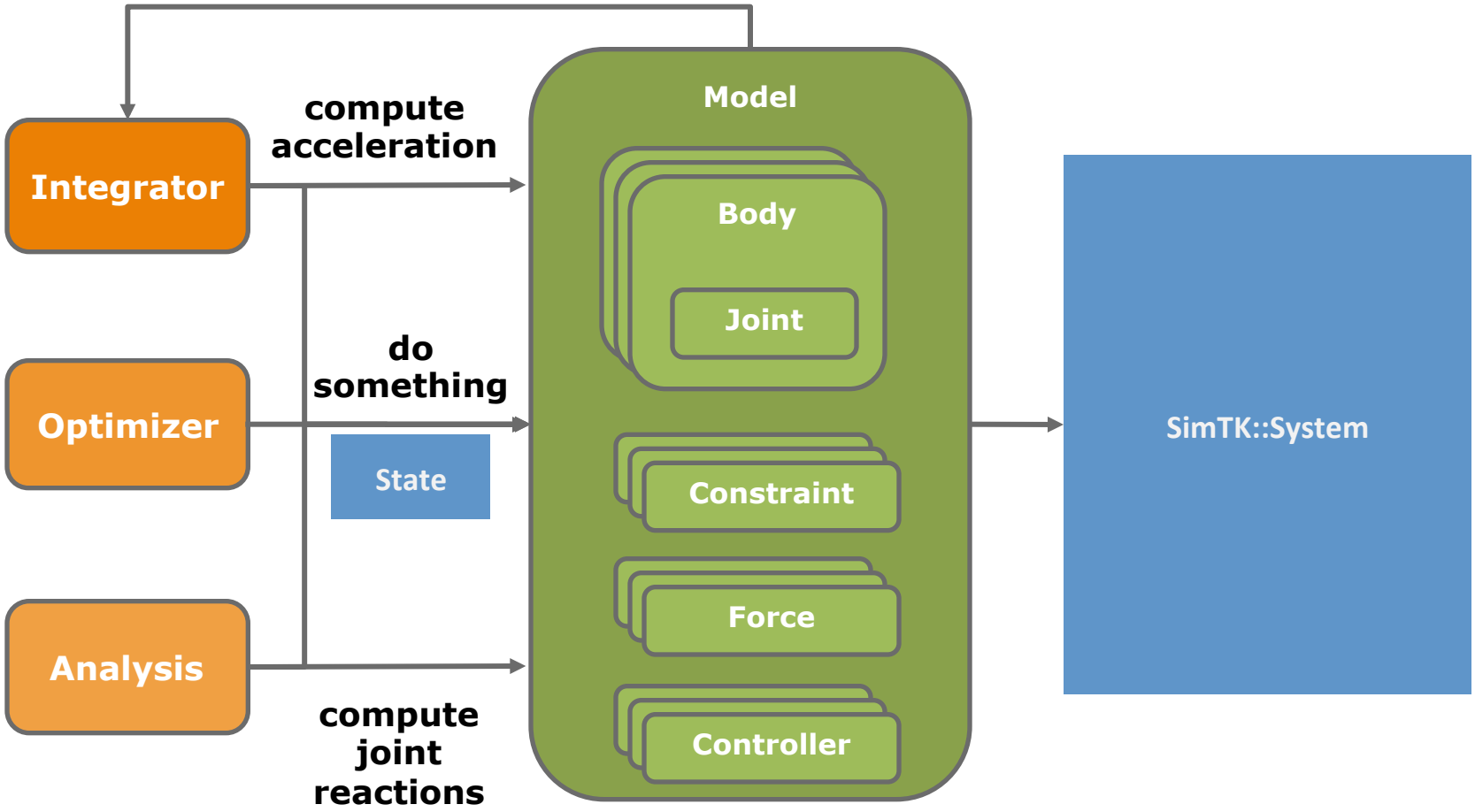
Organization vs. Computation

An OpenSim model and its components encapsulate properties of the physical system (mass, inertia, strength, etc …) and know how to represent themselves in an underlying computational system (equations) that can be solved.



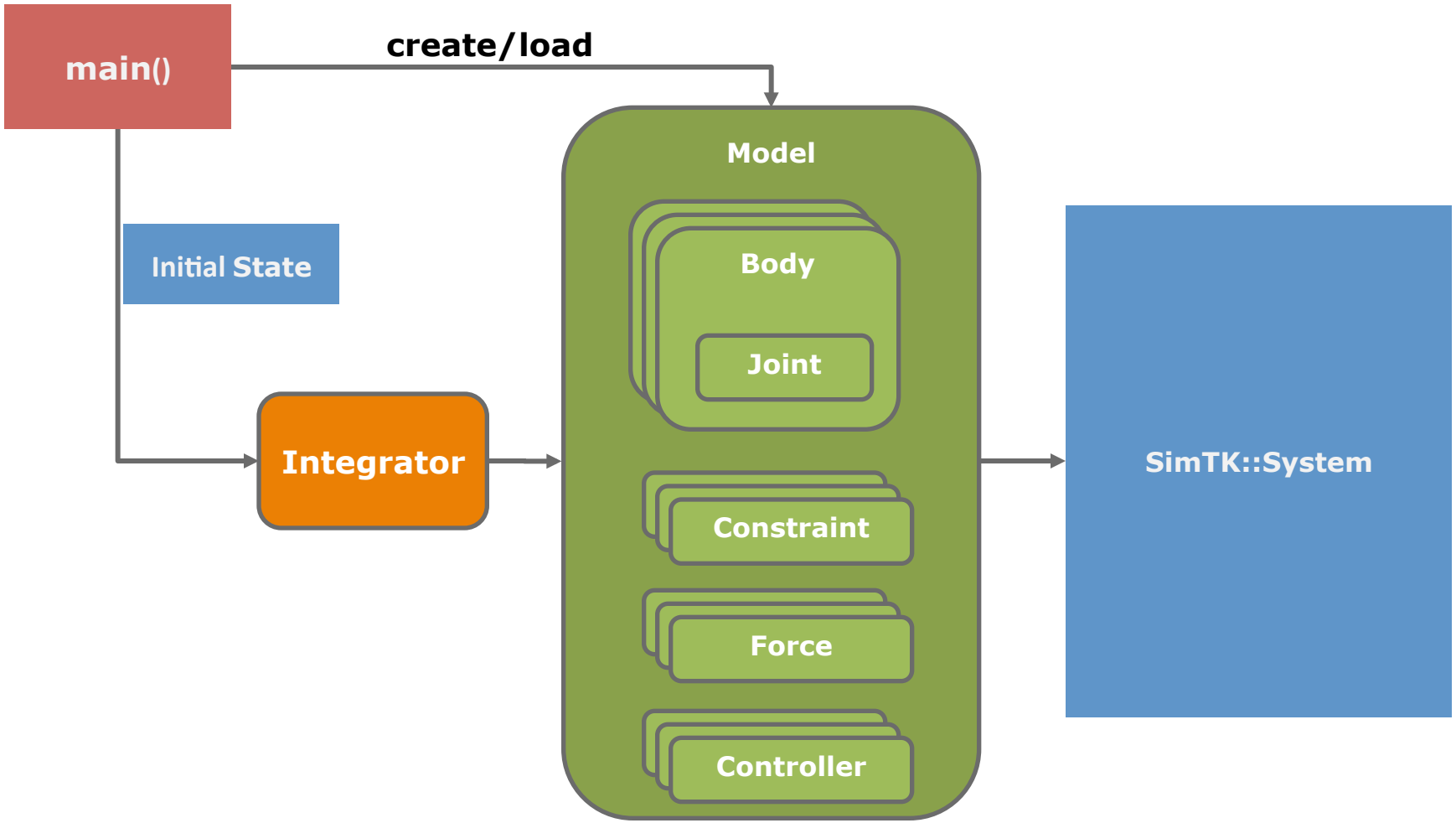
Objects that Make a Model "Do Something"

A model is called by higher level objects to perform (calculate something) and provide the model with a state.



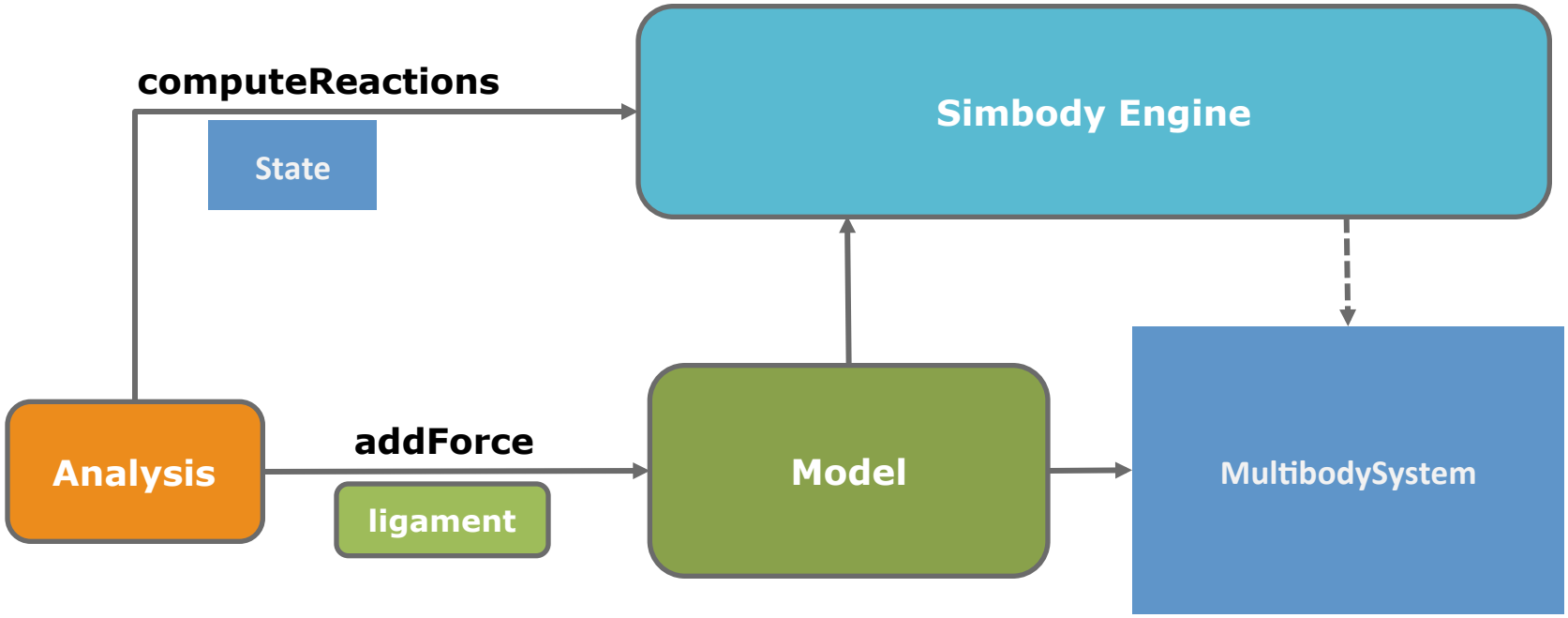
Your Program Creates & Commands Objects

You can define a main program to create a model and its manager and then execute a simulation.

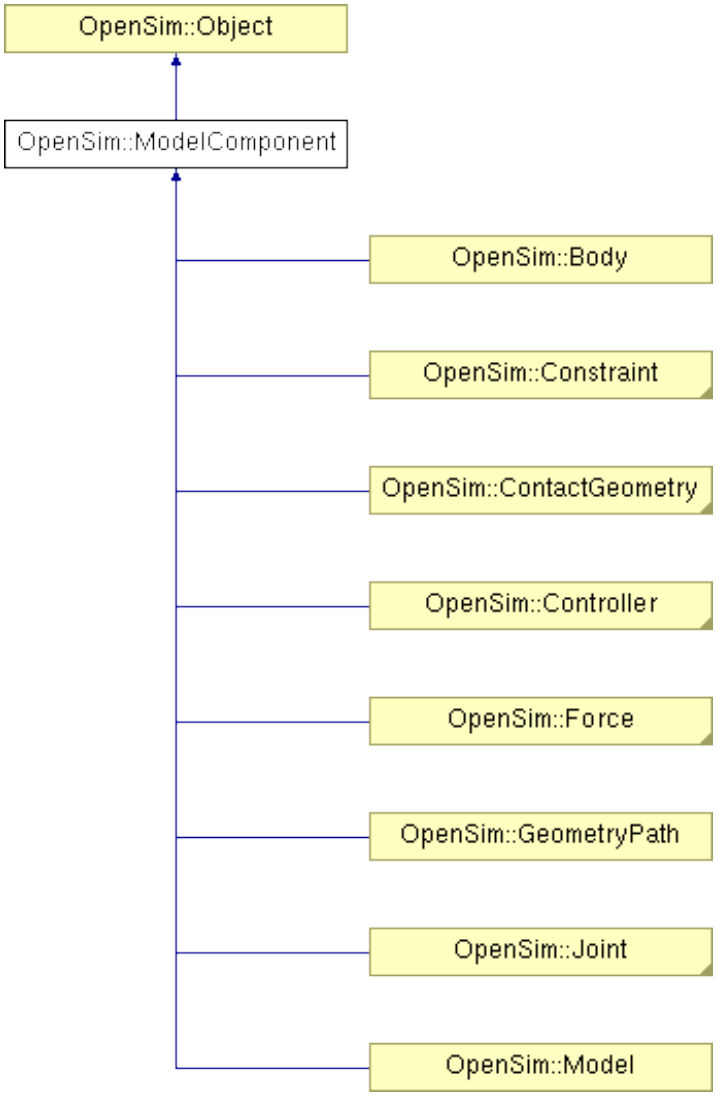


The SimbodyEngine (DynamicsEngine)

The SimbodyEngine is a convenience interface to the computational multibody dynamics system (Simbody) specified by the OpenSim Model.



OpenSim API Defined by Indidividual Classes



All components of an OpenSim model derive from the ModelComponent class and implement **`createSystem`**()

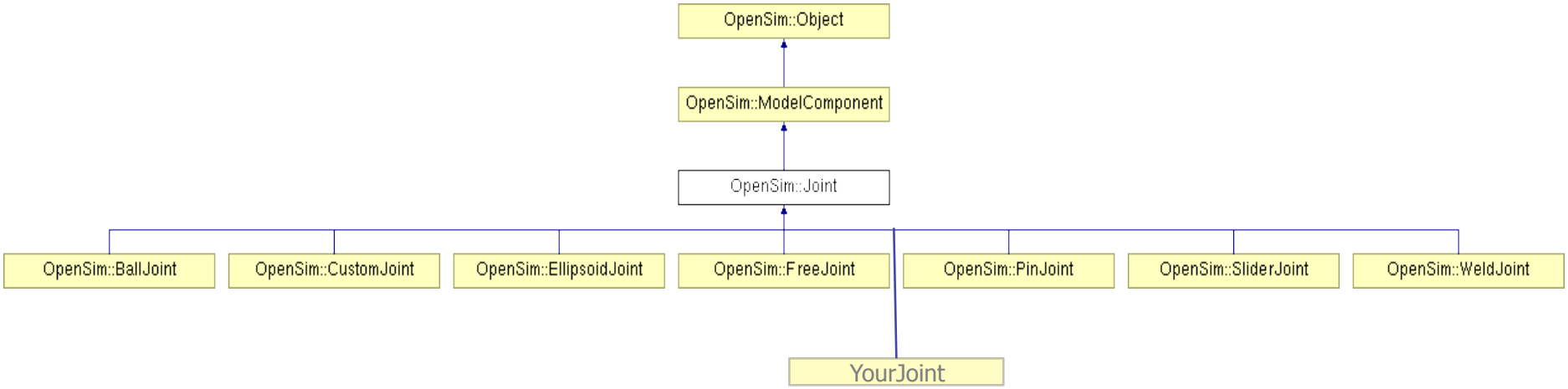
ModelComponent's can add its own state variables to the system State

ModelComponent is an OpenSim::Object, which supports reading and writing properties to/from XML

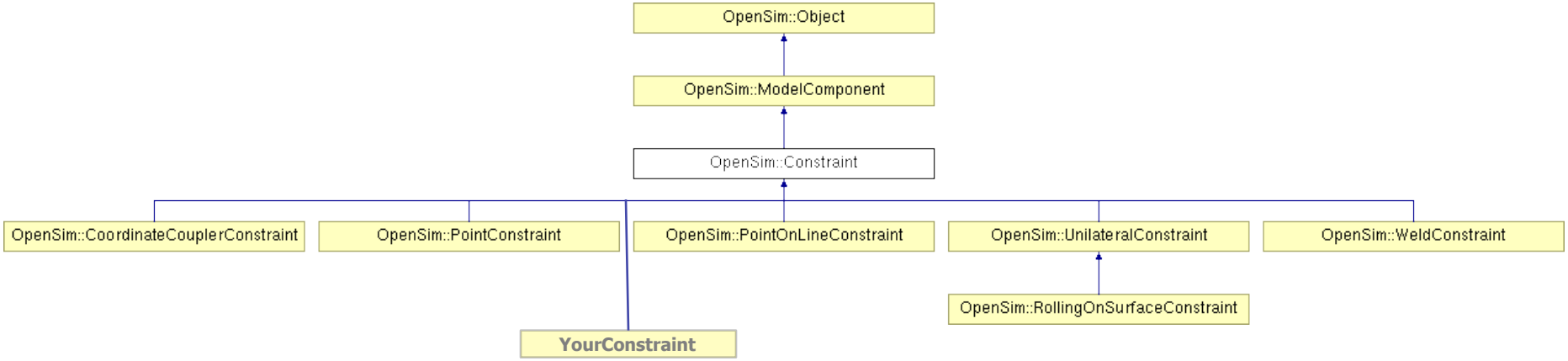
Object Hierarchy and Class Definitions available as Doxygen documentation:

<OpenSim2.3_Install>\sdk\doc\html\index.html

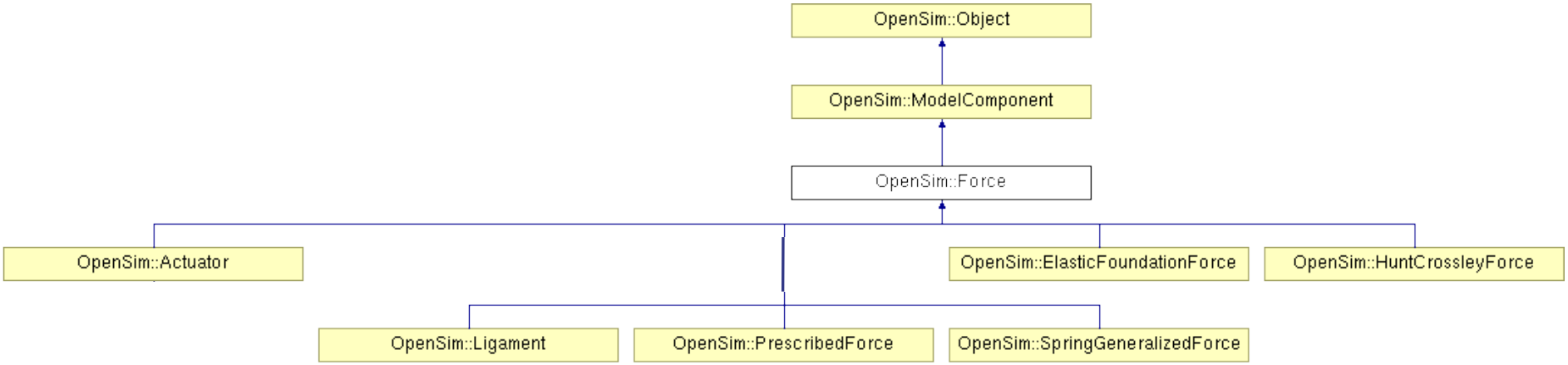
Specific Joint Behavior by Subclasses



Specific Constraint Behavior by Subclasses



Specific Force Behavior by Subclasses



A force can be dependent on and add state values to the system state.

An actuator is a force that is also dependent on control(s).

Controllers provide control values to actuators.

Getting More Information

1. Handout:

- previewing experimental data
- more tips and tricks

2. General Information: User's Guide

- https://simtk.org/docman/view.php/91/1022/OpenSim_UsersGuide.pdf

3. Ask OpenSim executable(s)

- Try:

```
ik -PropertyInfo (-PI) BushingForce
```

4. Application Programming Interface (API) examples

- <OpenSim_InstallDir>/sdk/APIExamples

5. API of underlying OpenSim (C++) classes (Doxygen)

- <OpenSim_InstallDir>/sdk/doc/index.html