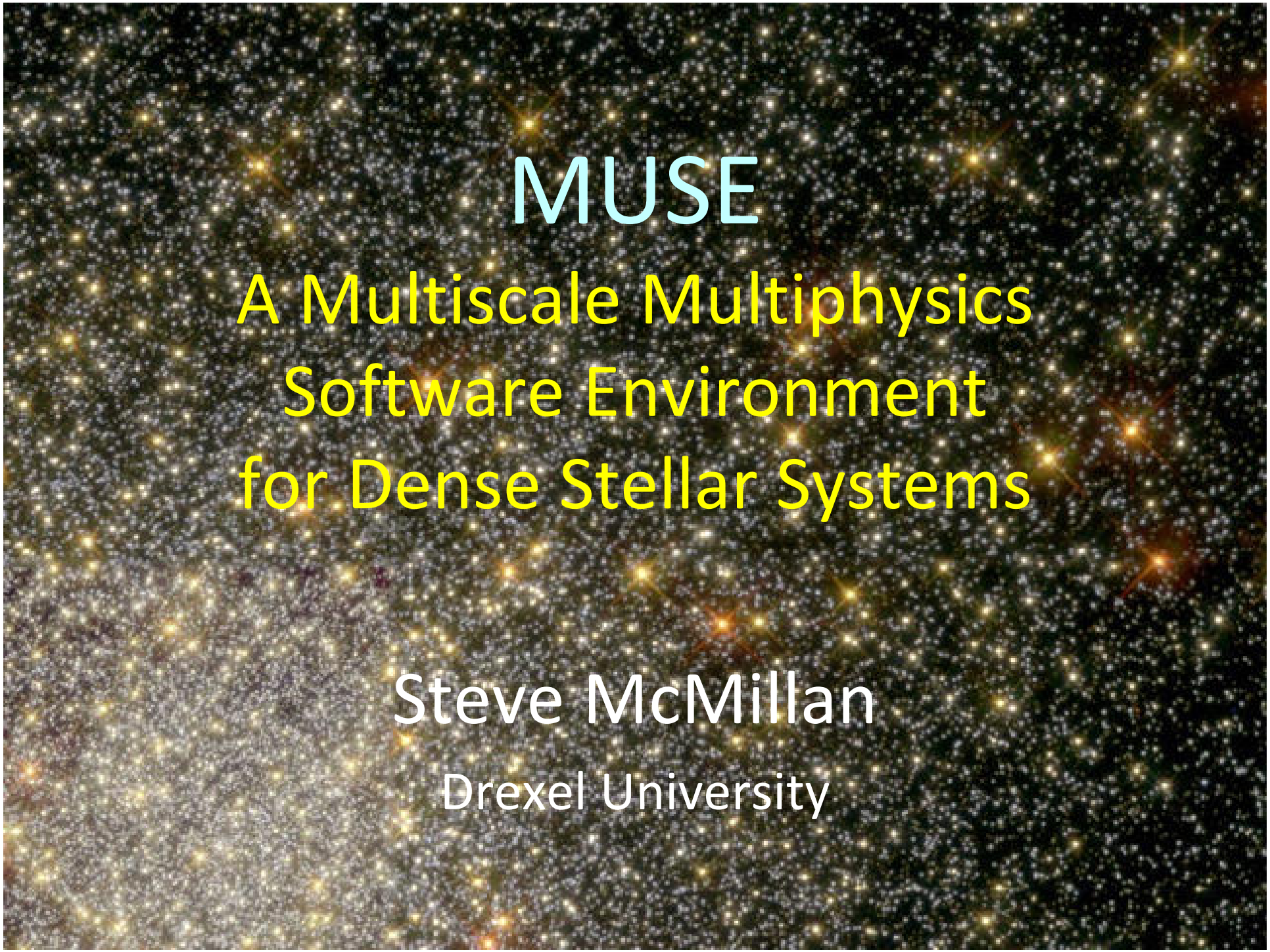

MUSE
A Multiscale Multiphysics
Software Environment
for Dense Stellar Systems
Steve McMillan
Drexel University


MUSE
collaborators
Evert Glebbeek
Stefan Harfst
Douglas Heggie
James Lombardi
Simon Portegies Zwart

Overview

- dense stellar systems
- software issues
- the MUSE software framework
- examples
  - mixed dynamical simulations
  - dynamics and stellar evolution (×2)

Dense Stellar Systems

- star forming regions
- young star clusters
- globular clusters
- nuclear star clusters
- galactic nuclei

Dense Stellar Systems

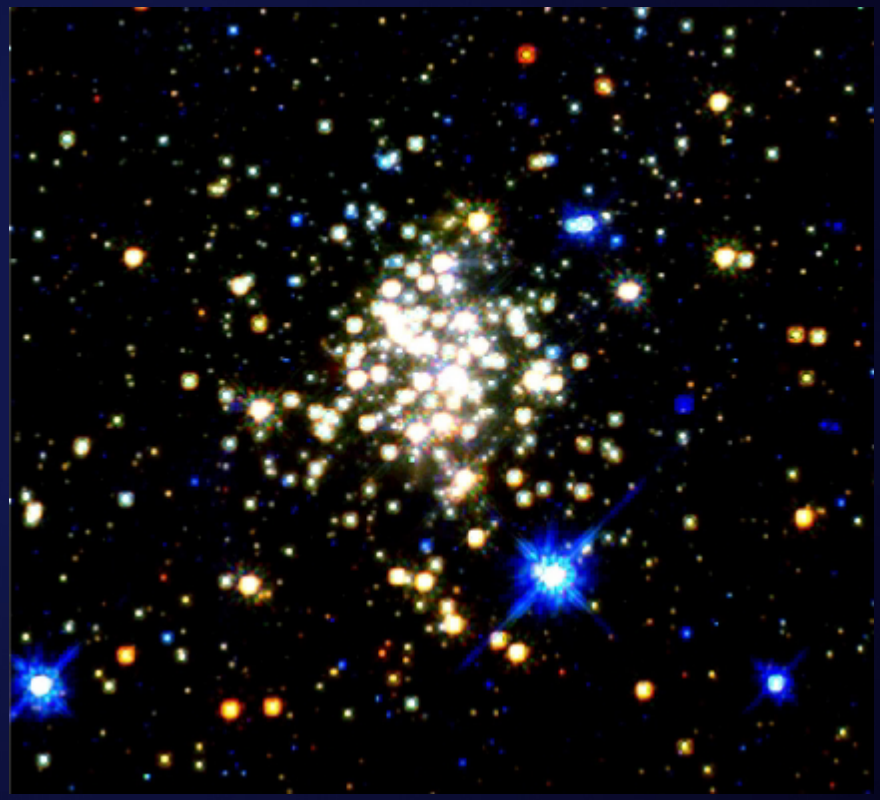
- star forming regions
- young star clusters
- globular clusters
- nuclear star clusters
- galactic nuclei

Dense Stellar Systems

- star forming regions
- young star clusters
- globular clusters
- nuclear star clusters
- galactic nuclei

Dense Stellar Systems

- star forming regions
- young star clusters
- globular clusters
- nuclear star clusters
- galactic nuclei



Dense Stellar Systems

- star forming regions
- young star clusters
- globular clusters
- nuclear star clusters
- galactic nuclei

Astrophysical Issues

- (dynamics leads to) interactions among stars in dense environments
- evolution driven by binaries, multiples, mass transfer, stellar evolution and physical stellar collisions
- stellar collisions force interplay among physical phenomena…

Astrophysical Issues

- (dynamics leads to) interactions among stars in dense environments
- evolution driven by binaries, multiples, mass transfer, stellar evolution and physical stellar collisions
- stellar collisions force interplay among physical phenomena…
- …and among astrophysicists

AMNH, New York, June 2002

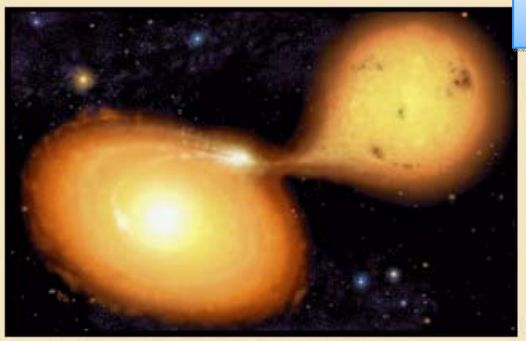
MODEST

http://www.manybody.org/modest/

Apple Google Maps YouTube Yahoo! News Popular

WG1 | WG2 | WG3 | WG4 | WG5 | WG6 | WG7 | WG8 | WG9 | WG10
MODEST-1 | MODEST-2 | MODEST-3 | MODEST-4 | MODEST-5 | MODEST-6 | MODEST-7 | MODEST-8 | MODEST-9

manybody.org/modest



MODEST

MOdeling DEnse STellar systems

- FAQs
- Primer
- Workshops
- Working Groups
- Mailing List
- Jobs Page
- Projects
- Other Links

Most stars in most galaxies will never experience a collision or even a close encounter with another star. Typical collision time scales in the solar neighborhood of the Milky Way galaxy exceed the age of the universe by many orders of magnitude, so physical stellar interactions are extremely rare. However, in some parts of the universe -- in galactic nuclei and some star clusters -- circumstances have conspired to create conditions in which physical collisions between stars are commonplace events. Such *dense stellar systems* stand at the interface between stellar dynamics and stellar evolution. Often owing their existence to purely dynamical processes, dense stellar systems offer wholly new channels for stars to evolve, allowing the formation of stellar species completely inaccessible by standard stellar and binary evolutionary pathways.

MODEST Goals

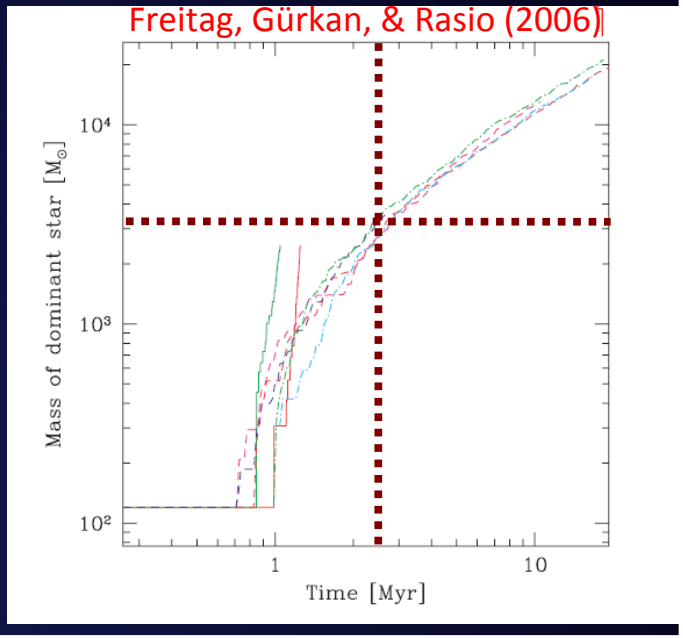
MODEST is a loosely knit initiative of various groups working in stellar dynamics, stellar evolution, and stellar hydrodynamics. Our aim is to provide a software framework for large-scale simulations of dense stellar systems, within which existing codes for dynamics, stellar evolution, and hydrodynamics can be easily coupled. While many of us have talked for years about combining `live' stellar evolution codes directly with N-body simulations, we have now reached a consensus between various groups about standards and interfaces, what is needed, and what is doable. On this web site we will provide up-to-date information about our activities, and pointers to various projects in progress, including coordination with numerous Virtual Observatory projects around the world.

Although neither we nor our project goals could possibly be described as modest, we use the name also to indicate that we intend only MODEST modifications of existing codes, in order

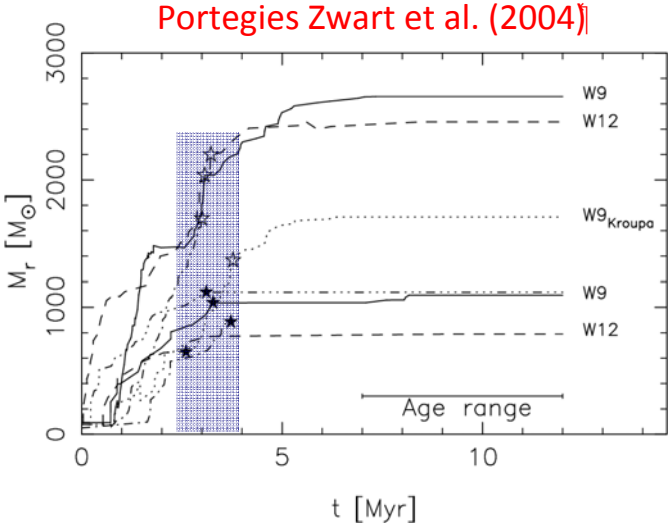
Runaway Stellar Mergers

- runaway mergers in clusters
  - dynamics → high densities
  - collisions → mergers
  - no mass loss (?)
    → supermassive stars
  - evolution (?)
    → intermediate-mass black holes
- tightly coupled multiphysics

Freitag, Gürkan, & Rasio (2006)



Portegies Zwart et al. (2004)



Computational Problems

- many physical processes
  - stellar dynamics on large and small scales
  - stellar and binary evolution
  - stellar encounters and collisions
  - internal gas dynamics on many scales
  - radiative transfer
  - …
- can't easily be separated

The State of the Art

- NBODY4/6/6++ + BSE + sticky spheres

  Aarseth, Hurley, Tout, Spurzem,…

- kira + seba + sticky spheres

  McMillan, Portegies Zwart, Hut, Makino

- MC + startrack/BSE + sticky spheres

  Rasio, Fregeau, Gurkan, Belczynski, Kalogera

  – in all cases: SPH/MMAS after the fact (Lombardi, Gaburov)

“Kitchen Sink” Codes

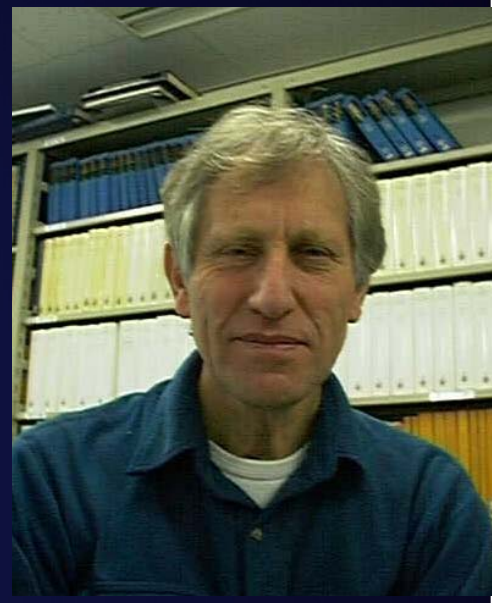
- very successful for MODEST-style problems
- monolithic design
- limited physics menu
  - detailed dynamics
  - approximate stellar evolution
  - semi-analytic/heuristic binary evolution
  - cartoon hydrodynamics
- hard to expand functionality

Software

- dense stellar systems combine related fields that have traditionally been pursued independently
- multiphysics problems, code integration essential
- don't want to reinvent the wheel
- legacy code base
- range of programming styles

Software

- dense stellar systems combine related fields that have traditionally been pursued independently
- multiphysics problems, code integration essential
- don't want to reinvent the wheel
- legacy code base
- range of programming styles



MUSE Design goals

- software framework connecting astro modules
  - address inflexibility in kitchen sink codes
- interoperability: “plug and play”
  - enable code calibration and comparison
  - allow easy creation of new programs/projects
- don’t hard-wire legacy codes!
- don't mandate a programming style or language
- incorporate existing code by wrapping

MUSE?

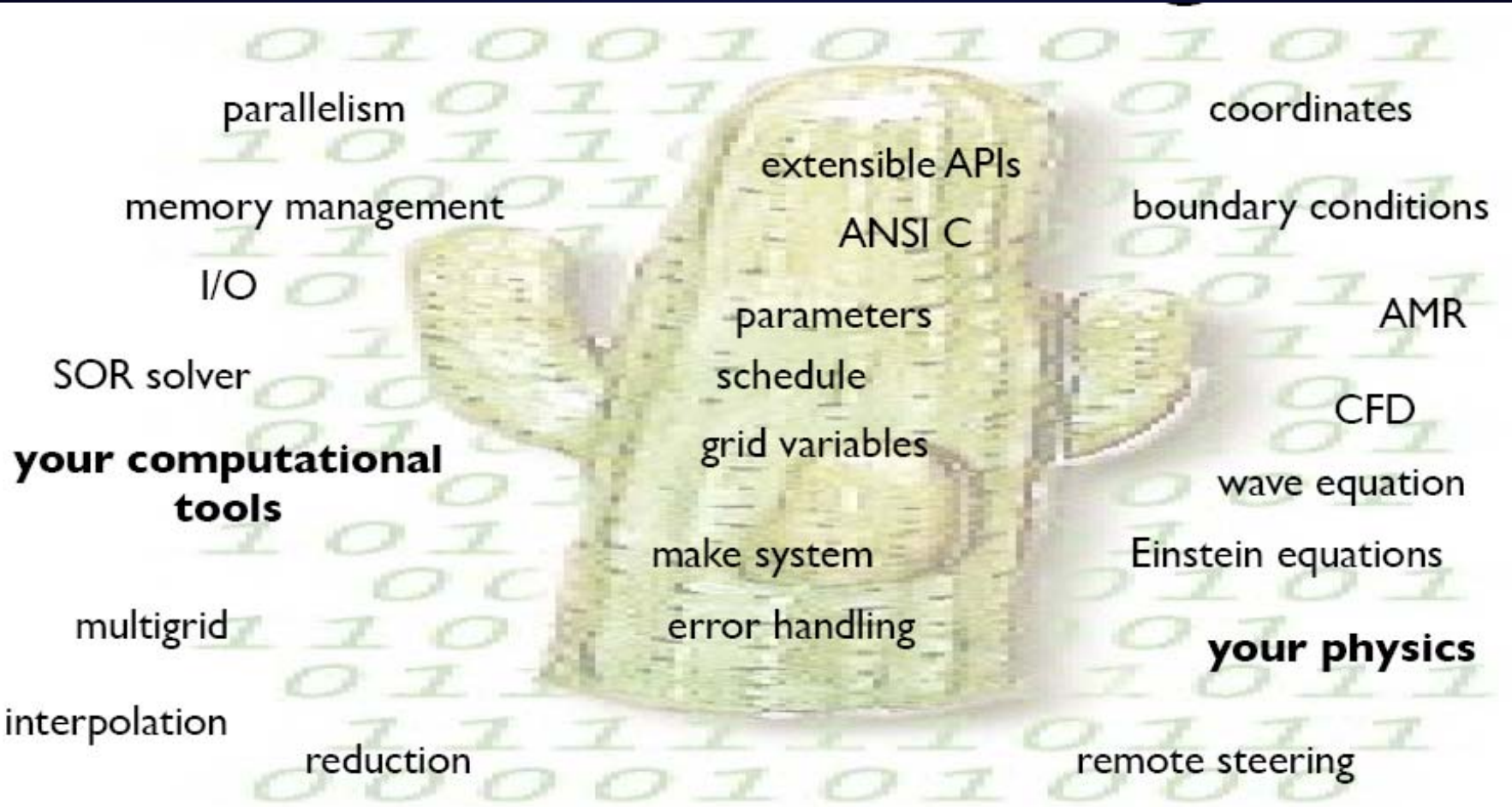
MUSE

- `http://muse.li`
- modules for stars, dynamics, collisions, etc.
- implemented as “black boxes” with wrappers
- well defined interfaces
- all modules provide prediction time scales, coordinated by “blind” scheduler
- use python as a top-level “glue” language

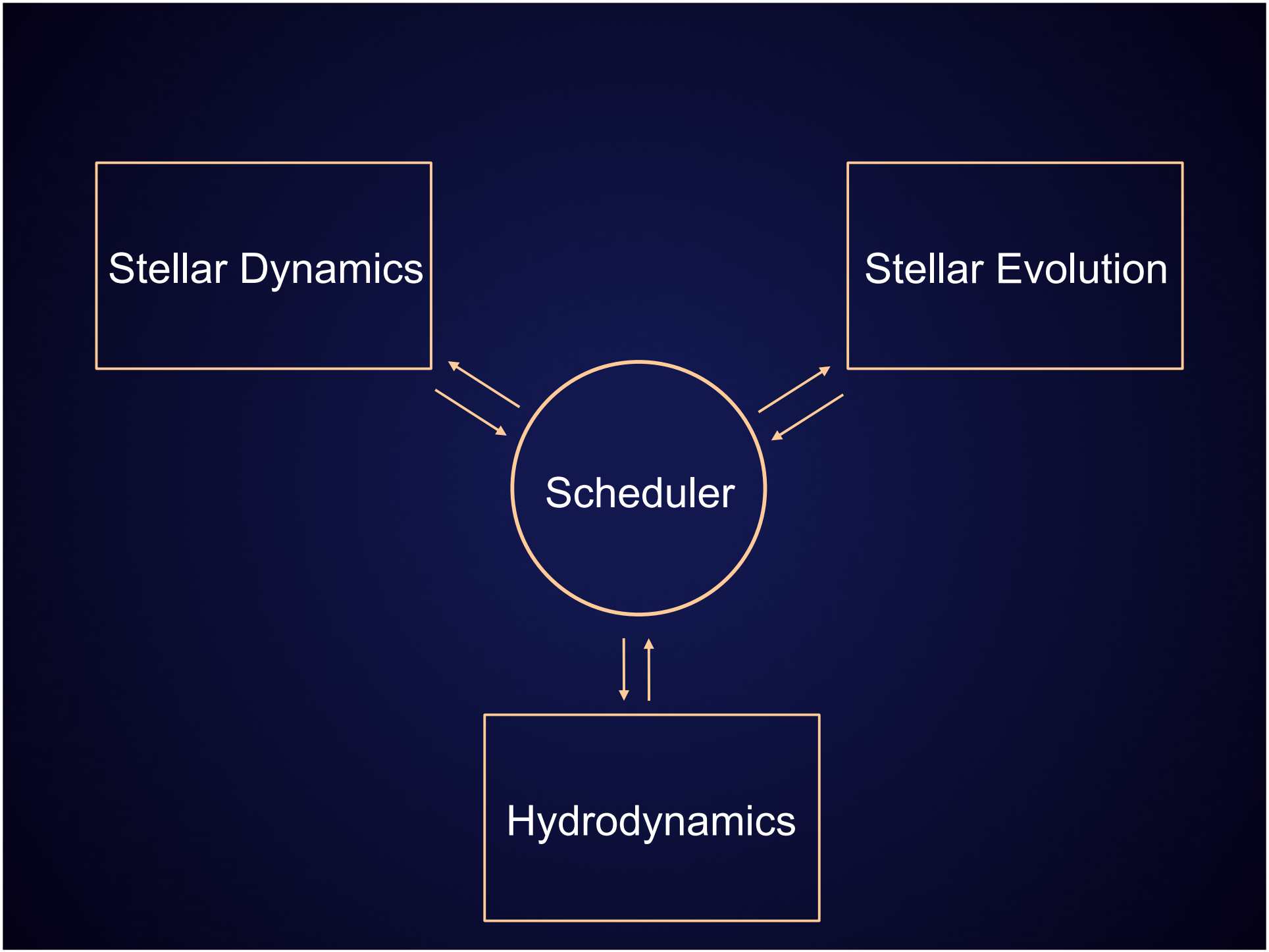
Python as a Glue Language

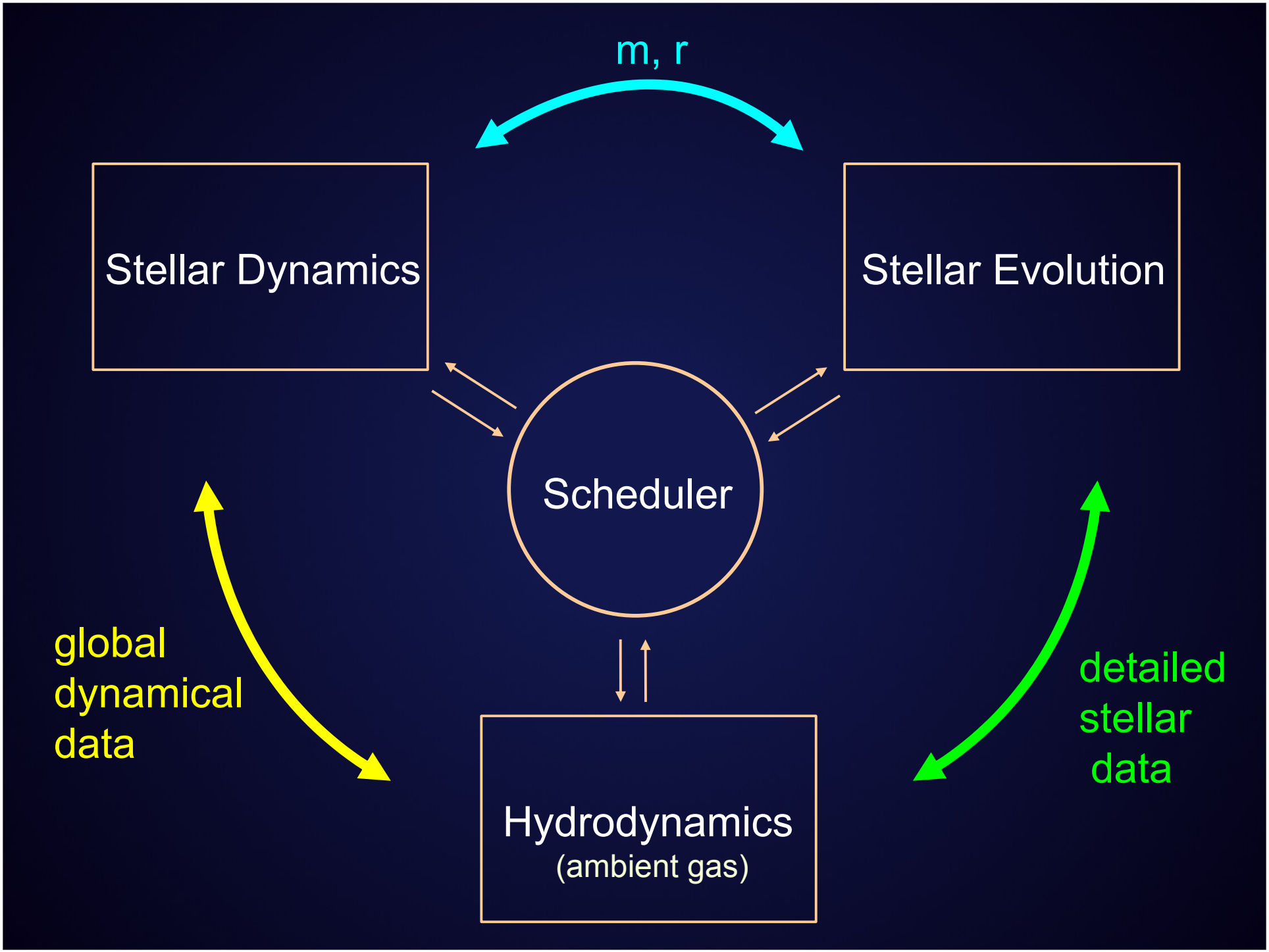
- flexible
- object oriented
- C, C++, f77, f90, f95 interfaces
- large user base
- many contributed modules
- numpy acceleration

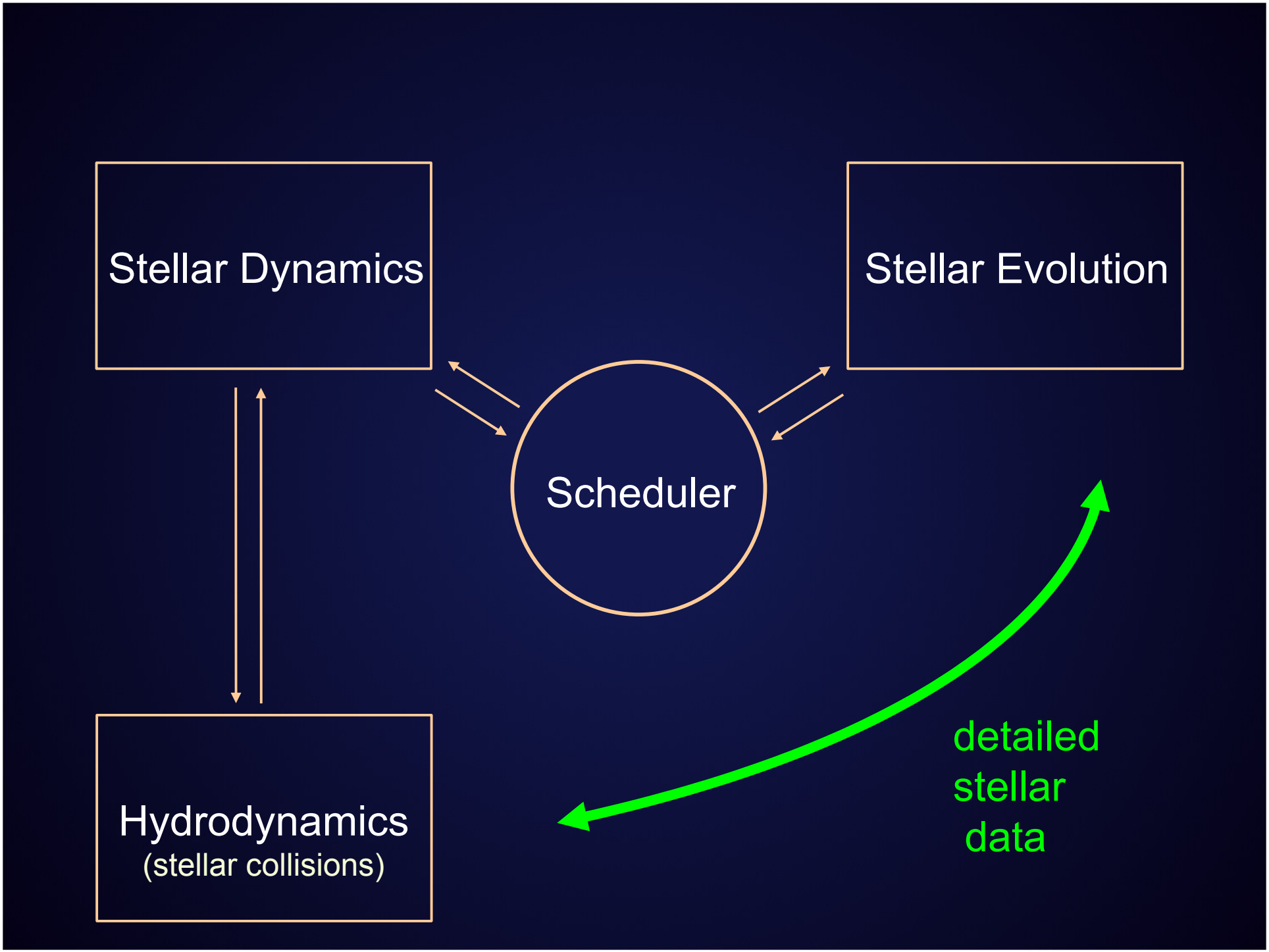
The CACTUS Project

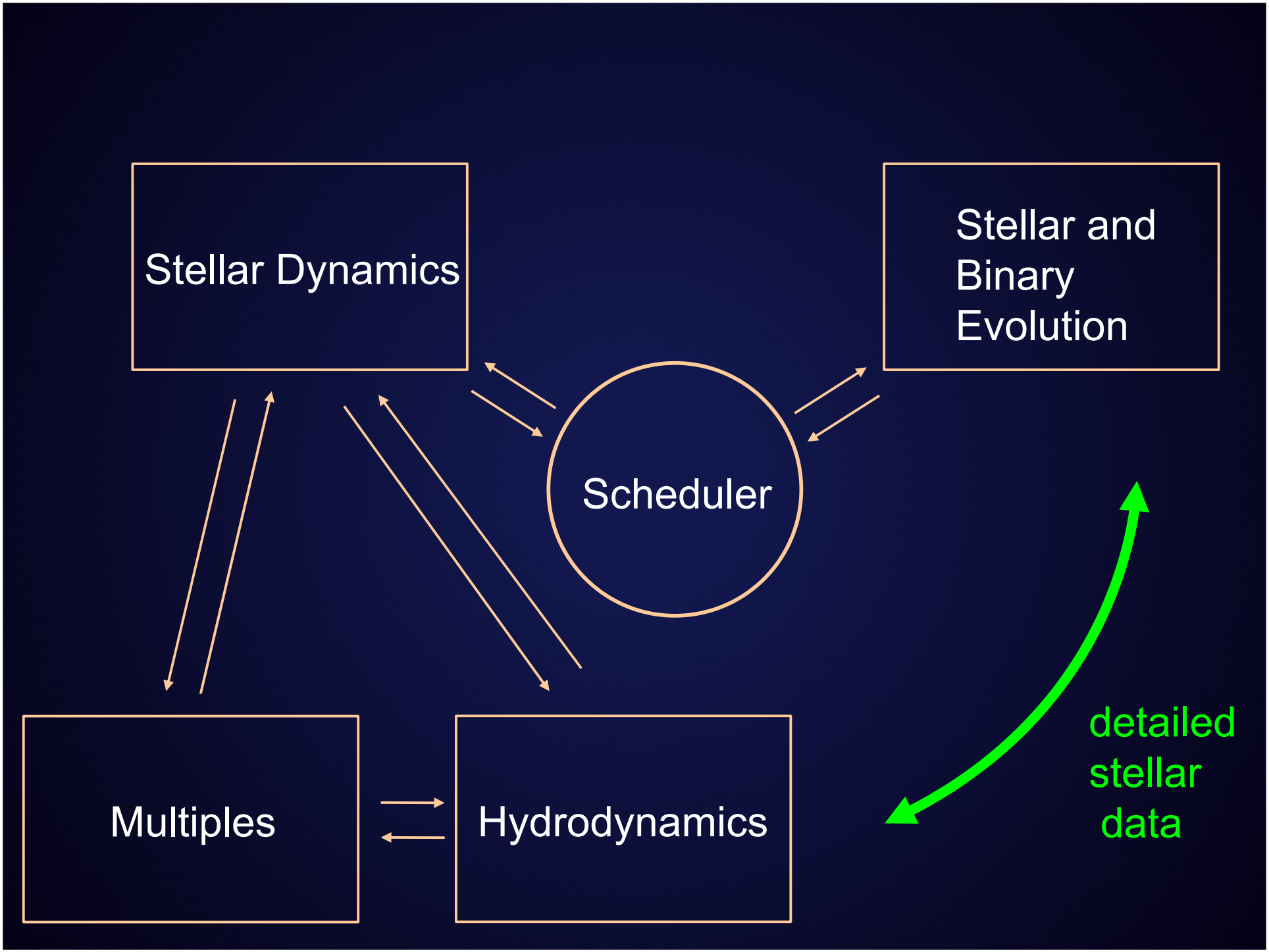


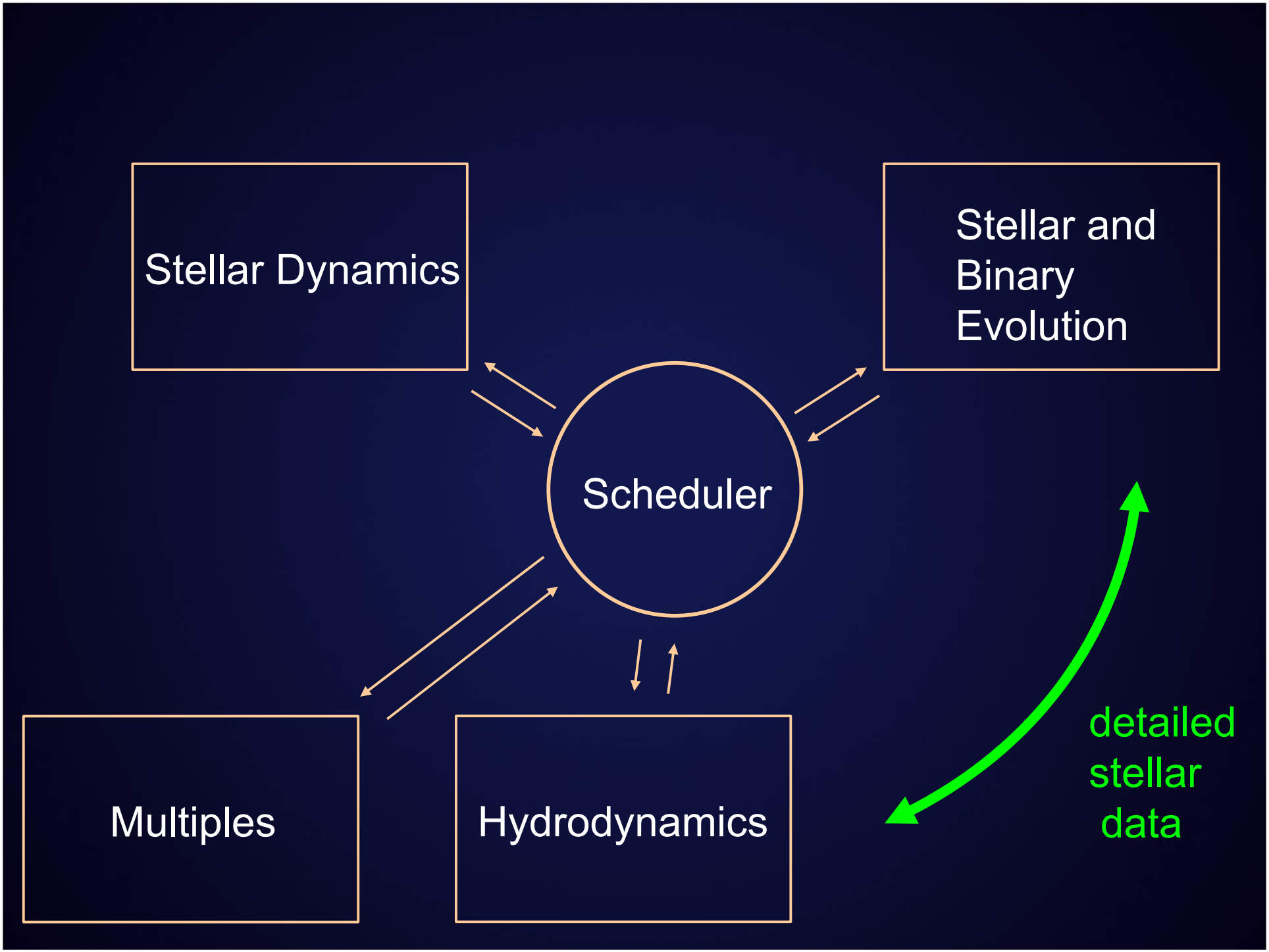
Core *flesh* with plug-in *thorns*


Stellar Dynamics
Stellar Evolution
Scheduler
Hydrodynamics


m, r
Stellar Dynamics
Stellar Evolution
Scheduler
global
dynamical
data
detailed
stellar
data
Hydrodynamics
(ambient gas)


Stellar Dynamics
Stellar Evolution
Scheduler
Hydrodynamics
(stellar collisions)
detailed
stellar
data


Stellar Dynamics
Stellar and Binary Evolution
Scheduler
Multiples
Hydrodynamics
detailed stellar data


Stellar Dynamics
Stellar and Binary Evolution
Scheduler
Multiples
Hydrodynamics
detailed stellar data

MUSE status

- currently have modules for
  - stellar dynamics (6)

    BHTree, CMC, hermite0, hermite1, nbody1h, φGRAPE
  - stellar evolution (3)

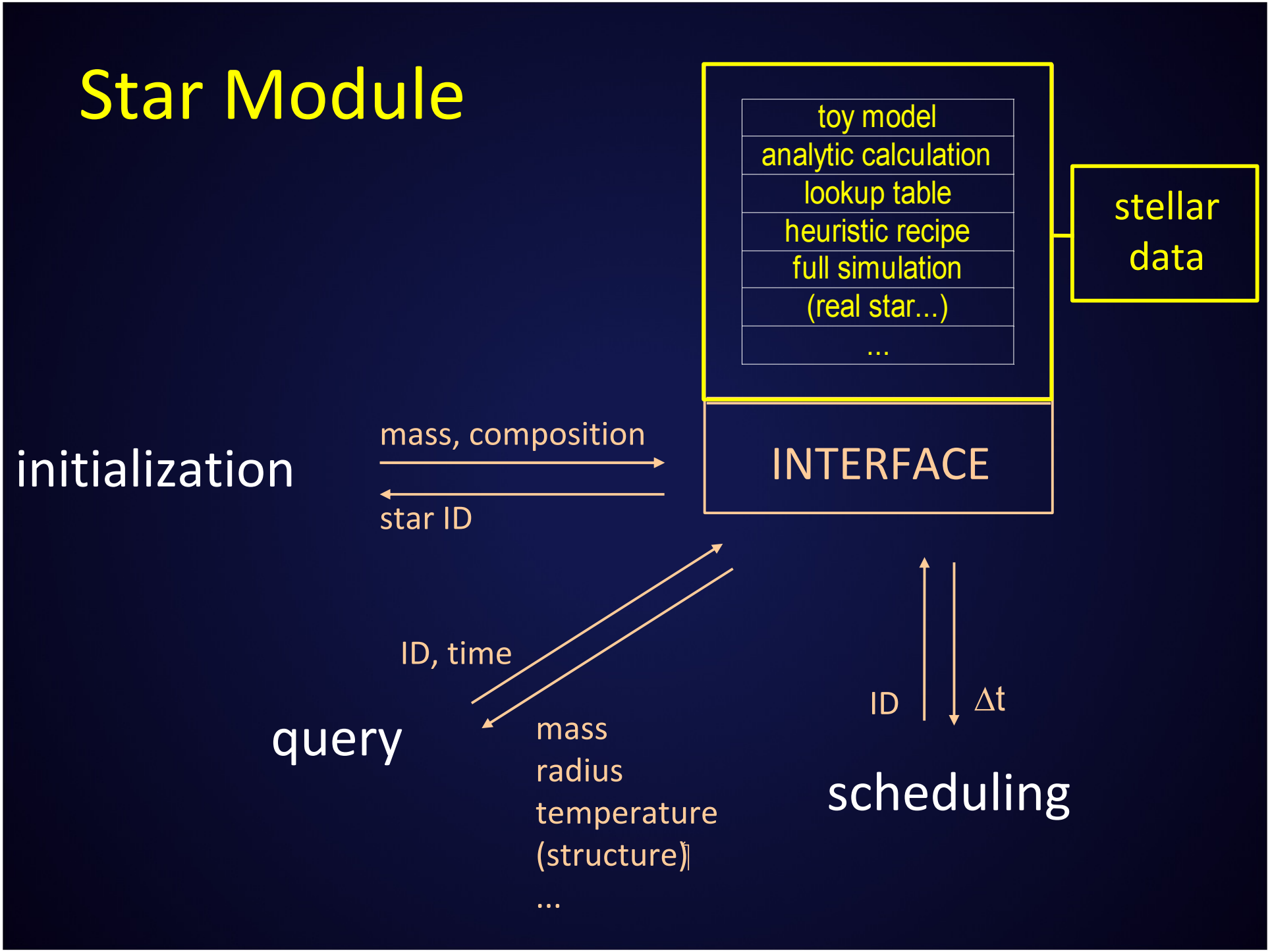
    EFT89, HPT00, EVTwin
  - stellar collisions (2)

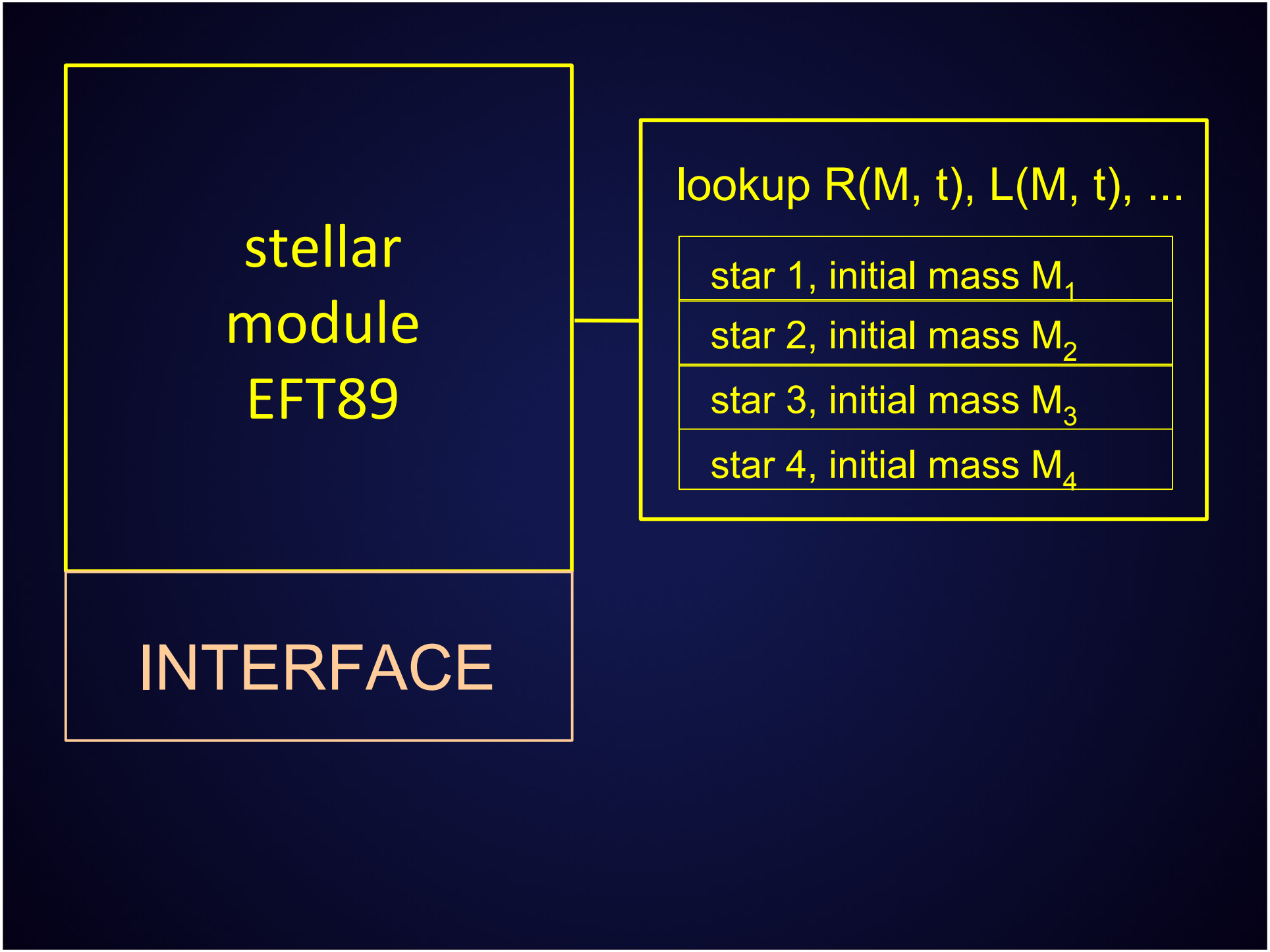
    sticky _spheres, MMAS
  - multiple encounters (1)

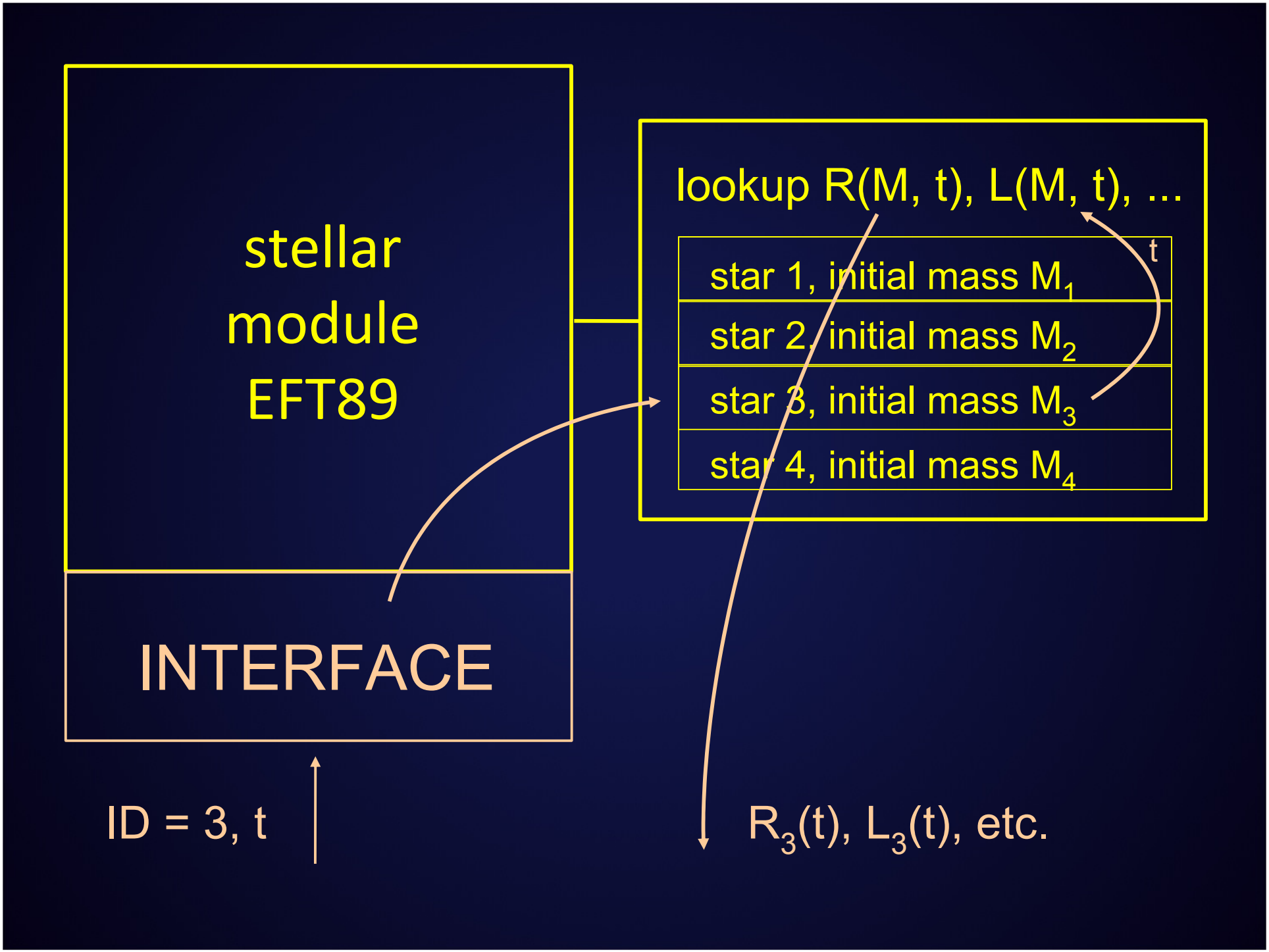
    smallN

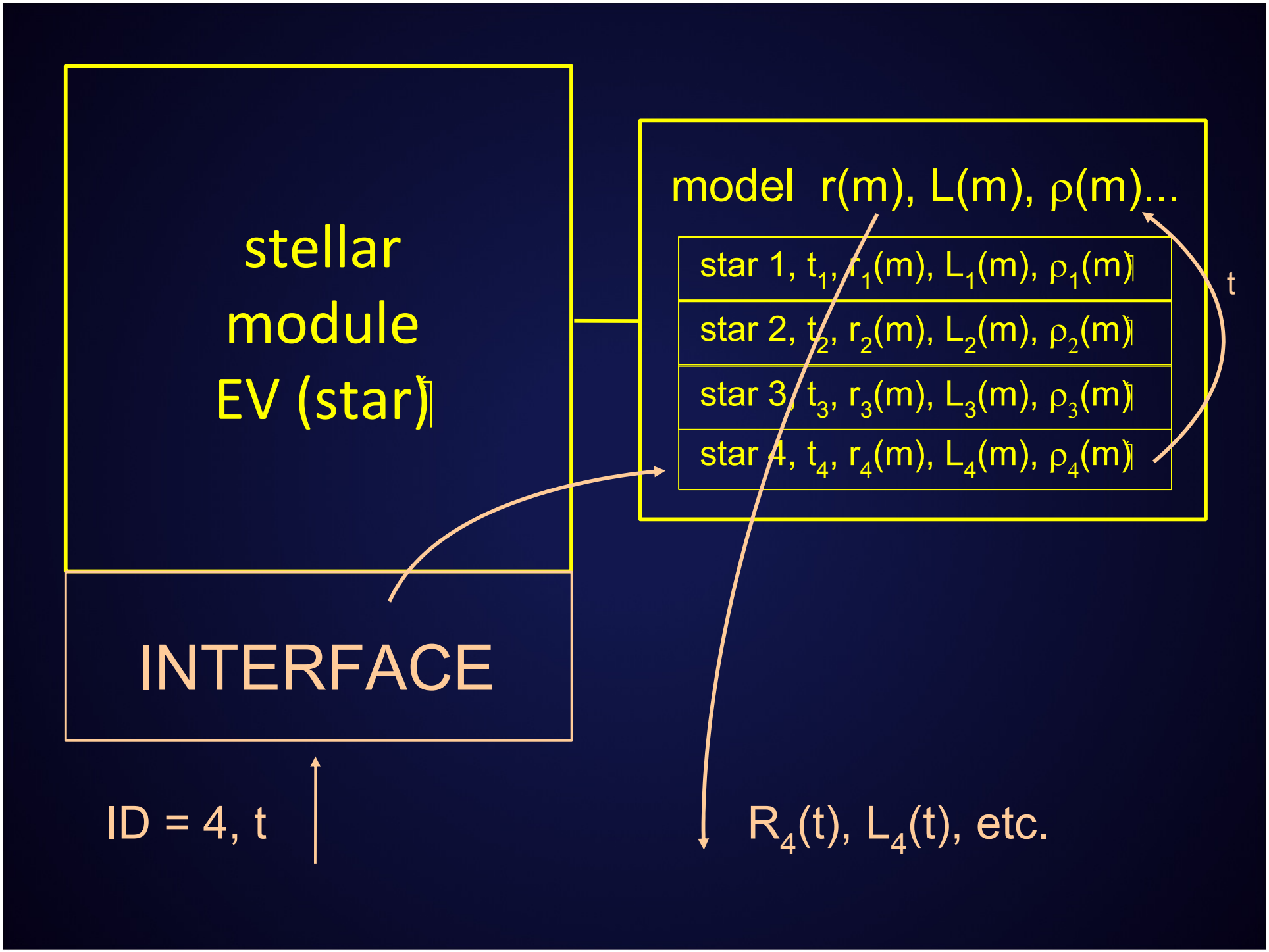
```
from gravity.hermite0.muse_dynamics import Hermite as dyn
from stellar.EFT89.muse_stellar import EFT89 as star
from collisions.sticky_spheres.muse_collisions import StickySpheres as coll
.
.   (initialization)
.
while time < t_max:

        time += dtime
        while dyn.get_time() < time:
                id1 = dyn.evolve(time)

                if id1 > 0:
                        id2 = dyn.find_colliding_secondary(id1)
                        evolve_stars(dyn.get_time())
                        collide_stellar_pair(id1, id2)

        evolve_stars(time)

print "end at t = ", time,  ", Nstars= ", star.get_number()
```

Star Module


toy model
analytic calculation
lookup table
heuristic recipe
full simulation
(real star...)
...
stellar data
INTERFACE
initialization
mass, composition
star ID
query
ID, time
mass
radius
temperature
(structure)
...
ID
Δt
scheduling


stellar
module
EFT89
INTERFACE
lookup R(M, t), L(M, t), ...
star 1, initial mass M_1
star 2, initial mass M_2
star 3, initial mass M_3
star 4, initial mass M_4


stellar
module
EFT89
INTERFACE
ID = 3, t
lookup R(M, t), L(M, t), ...
t
star 1, initial mass M1
star 2, initial mass M2
star 3, initial mass M3
star 4, initial mass M4
R3(t), L3(t), etc.


stellar
module
EV (star)
INTERFACE
ID = 4, t
model r(m), L(m), ρ(m)...
star 1, t1, r1(m), L1(m), ρ1(m)
star 2, t2, r2(m), L2(m), ρ2(m)
star 3, t3, r3(m), L3(m), ρ3(m)
star 4, t4, r4(m), L4(m), ρ4(m)
t
R4(t), L4(t), etc.

MUSE examples

- multiple dynamical integrators

  (Harfst 2008)

- dynamics and collisions

- dynamics and stars

  (McMillan & Glebbeek, 2008;
  Portegies Zwart et al 2008)

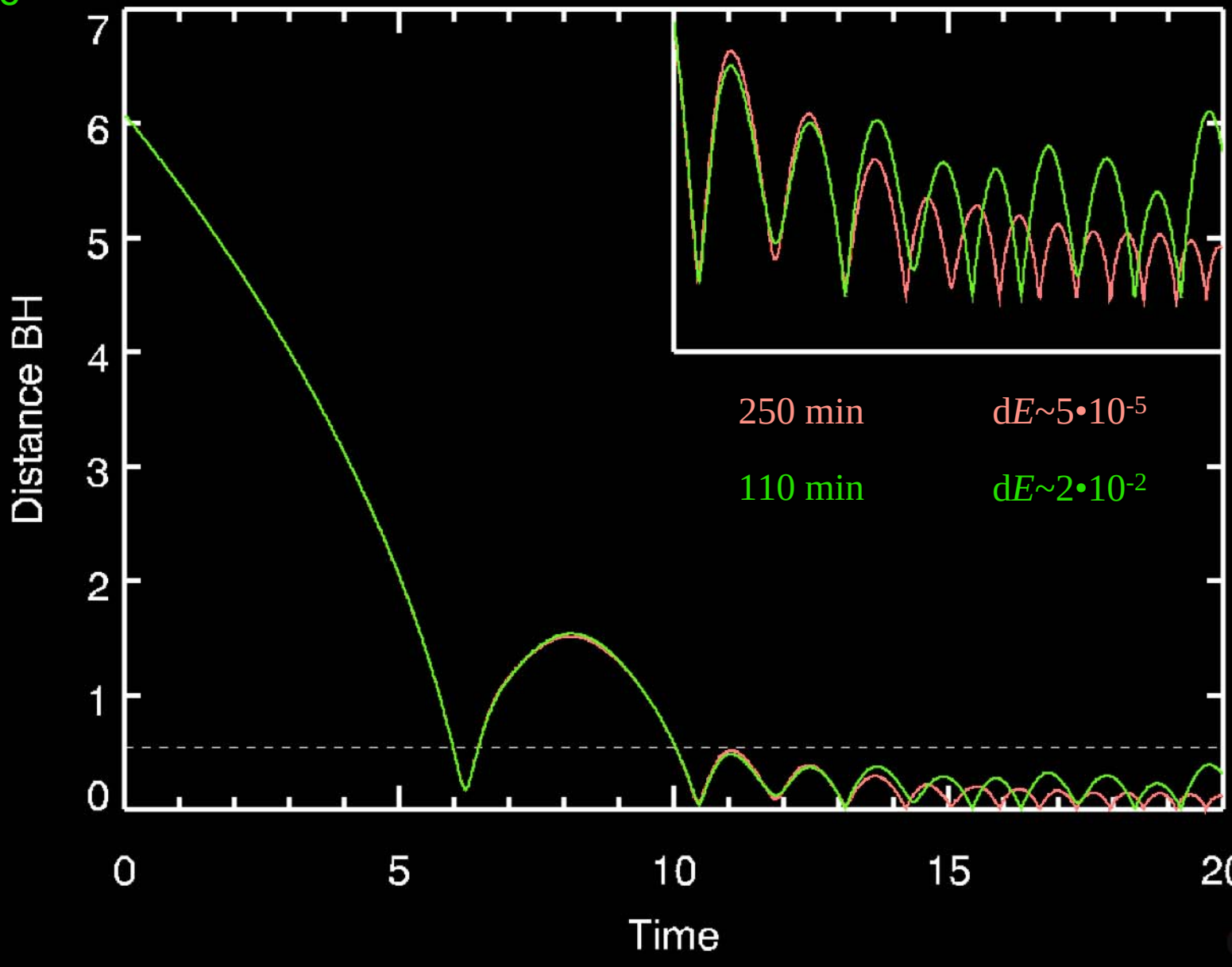
Switching integrators (Harfst 2008)

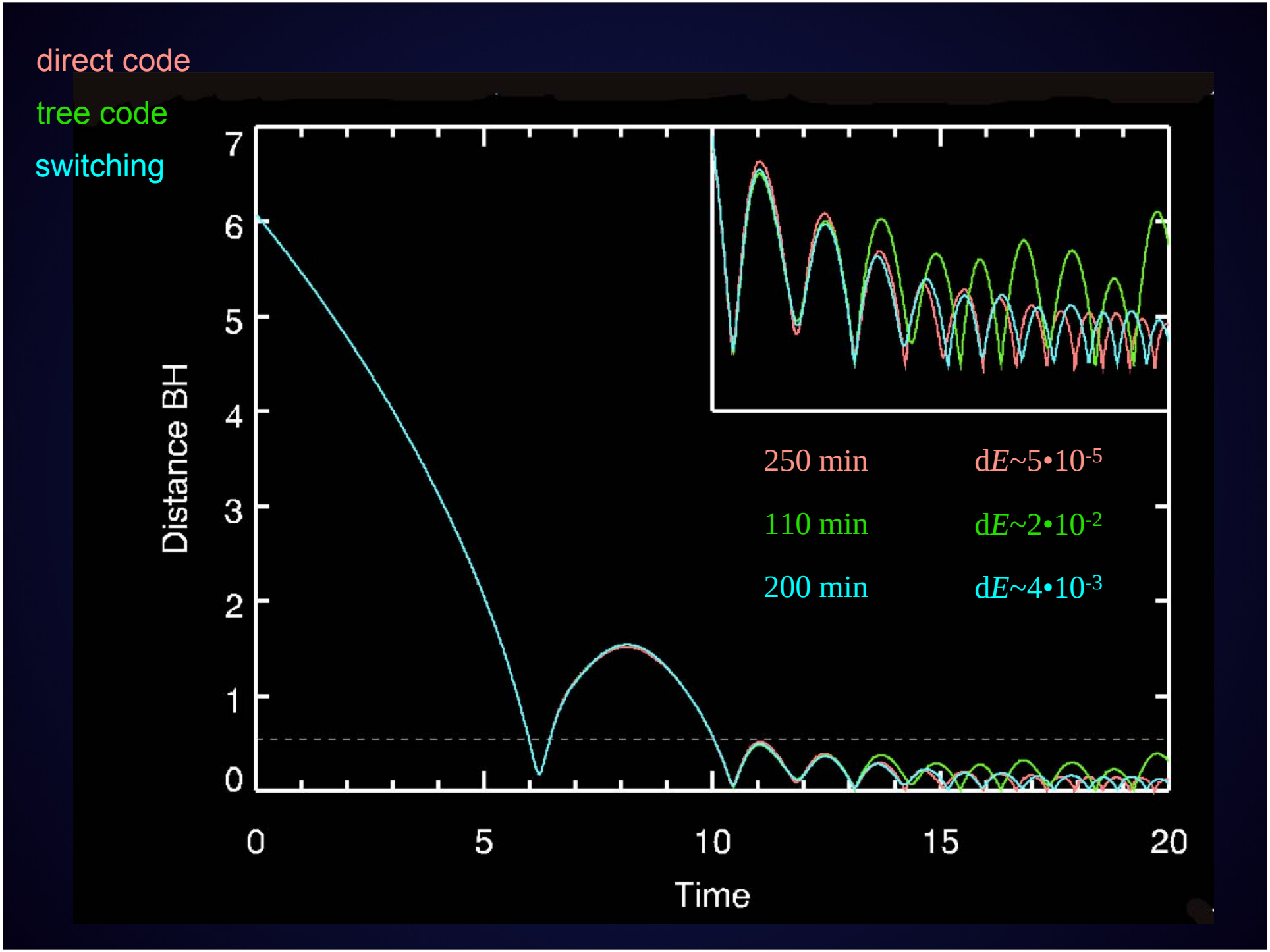
- merger of two galaxy/star clusters containing central black holes
  - two Plummer models, each N = 32k + 1
  - near head-on collision
  - runs with φGRAPE, BHTree, and switching from BHTree to φGRAPE when $d_{\mathrm{BH}} < 0.55$

    (both *N*-body codes use *GRAPE* acceleration)

direct code

tree code

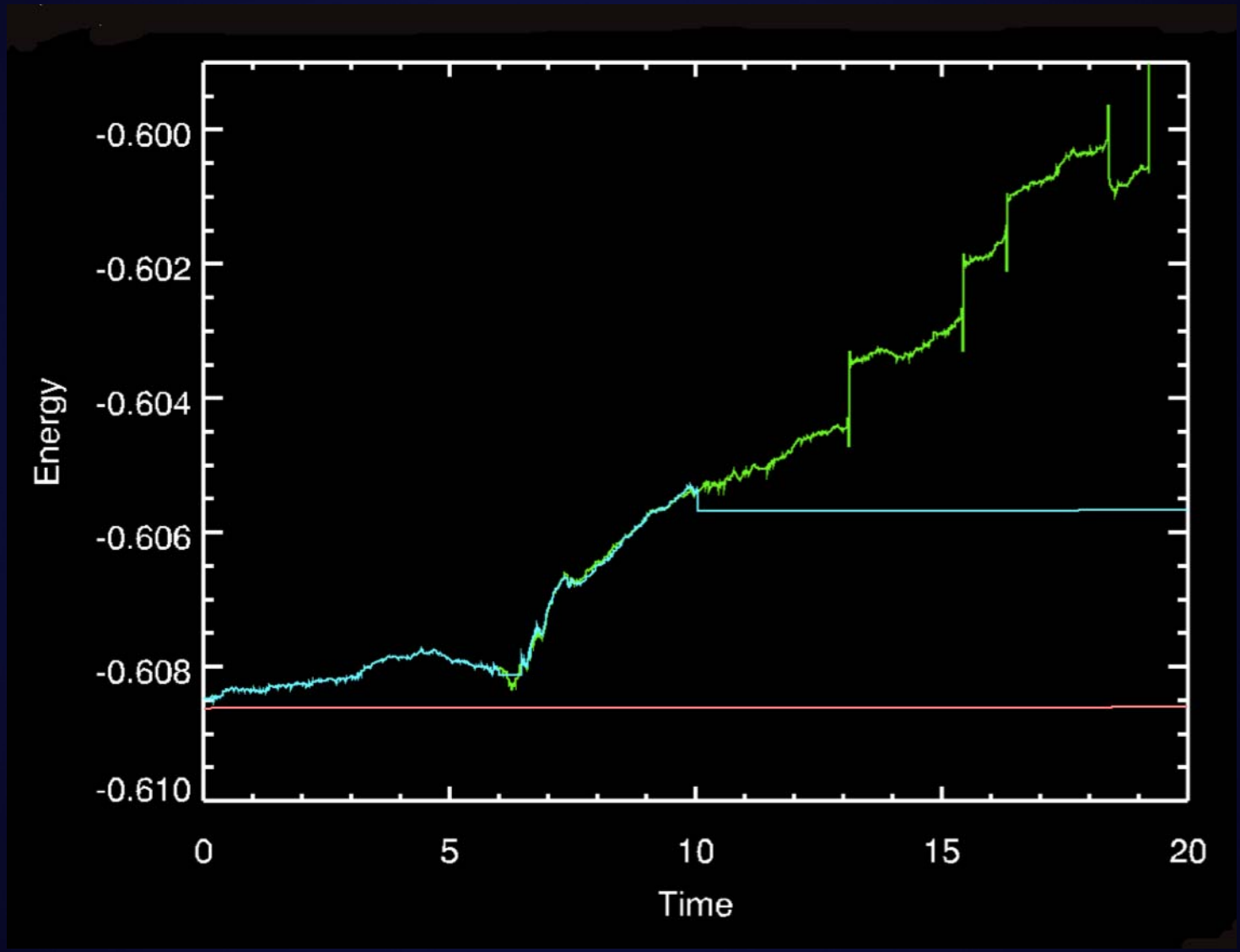



direct code
tree code
switching
Distance BH
Time
0
1
2
3
4
5
6
7
0
5
10
15
20
250 min
dE~5•10-5
110 min
dE~2•10-2
200 min
dE~4•10-3

direct code

tree code

switching

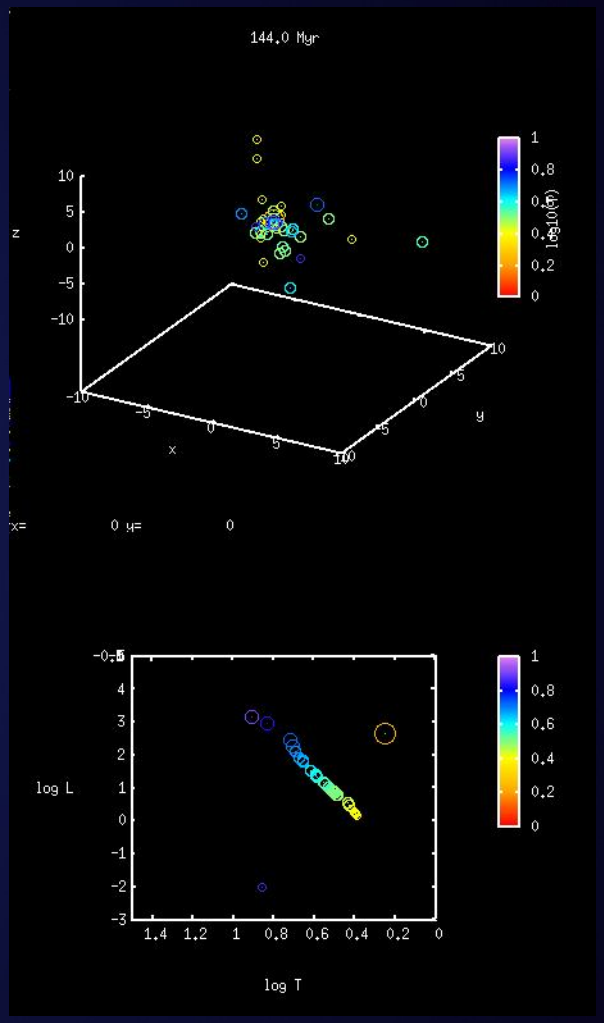


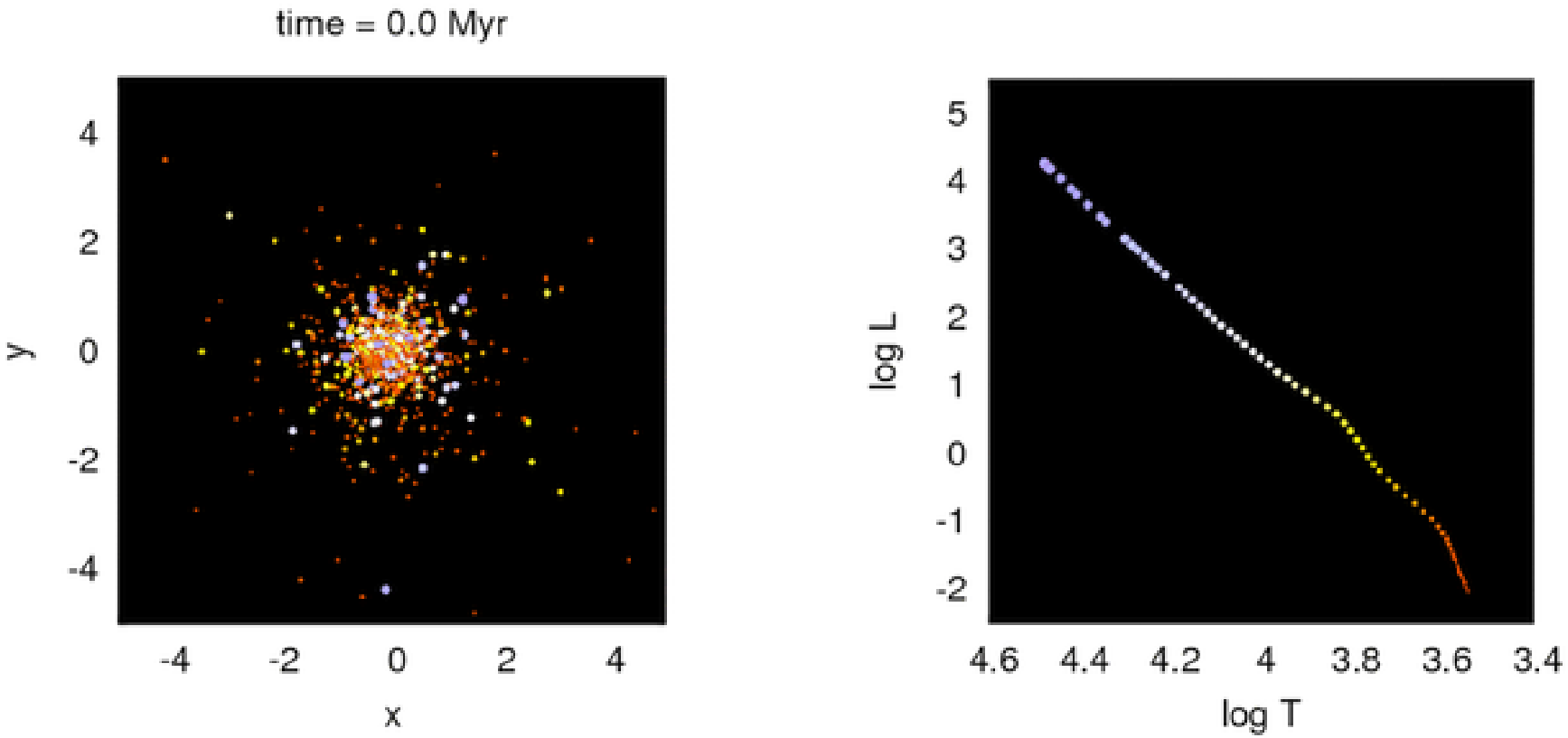
Stellar Evolution and Dynamics

1. simulations with dynamics and (simplified) stellar evolution and collisions

2. simulations with dynamics and (full) stellar evolution

```
import hermite0 as SD
import EFT89 as SE
import sticky_spheres as SC

.
. (initialisation)
.

while time <= t_end:

  time += dtime
  while SD.get_time()<time:
    id1 = SD.evolve(time)

    if id1 > 0:    # collision detected
      id2 = SD.find_colliding_secondary(id1)
      SE.evolve(SD.get_time())
      SC.collide_pair(id1,id2)

  SE.evolve(time)

  output()

print “end at t = ”,time,“ Nstars = ”,SE.get_number()
```




time = 0.0 Myr
y
x
4
2
0
-2
-4
-4
-2
0
2
4
log L
log T
5
4
3
2
1
0
-1
-2
4.6
4.4
4.2
4
3.8
3.6
3.4

Still to do

- dynamics of multiple interactions
- full integration of stellar collisions
- binary evolution
- star formation
- intracluster gas dynamics
- radiative transfer
- …

Conclusions

- MUSE:interchangeable modules, standardized interfaces
- enables experimentation and code comparison
- clean implementations of specific physics
  - parallel, GRAPE/GPU accelerated, grid enabled, etc.
- top-level code leverages python functionality
  - easy to add new modules

Conclusions

- MUSE:interchangeable modules, standardized interfaces
- enables experimentation and code comparison
- clean implementations of specific physics
  - parallel, GRAPE/GPU accelerated, grid enabled, etc.
- it won't work if no-one contributes!
  - consider larger-scale integration when writing code

Modest-8b Workshop

http://modesta.science.uva.nl/modest/modest8b/

Apple Google Maps YouTube Yahoo! News Popular

Welcome to the Modest-8b Workshop

A Multiscale Multiphysics Scientific workshop from 15 until 19 September 2008 in Amsterdam

SUMMARY

You are invited for the 5-day workshop in order to continue the development and discuss the future of MUSE - a Multi-scale Multi-physics Software Environment. The purpose of MUSE is to realistically model dense stellar systems such as young star clusters and galactic nuclei (see http://muse.li). Important steps had been taken over the last two years at previous workshops, most recently in Split, Croatia, in August 2007, where we accomplished to complete a working interface to connect a complete set of base codes. During this next meeting we plan to demonstrate the capabilities of MUSE by simulating a star cluster including stellar dynamics, stellar evolution as well as stellar collisions.

First Announcement Letter

Second Announcement Letter

Registration