LIO Institut des Origines de Lyon
UNIVERSITE DE LYON

CRAL
CENTRE DE RECHERCHE ASTROPHYSIQUE DE LYON

RASCAS

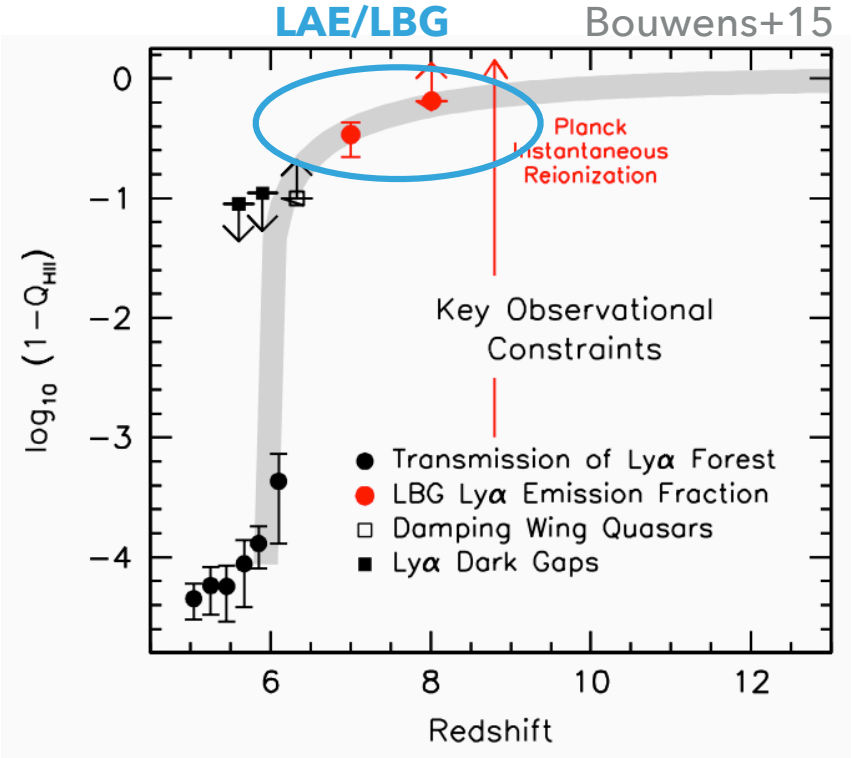
(aka MCLya v.2.0)

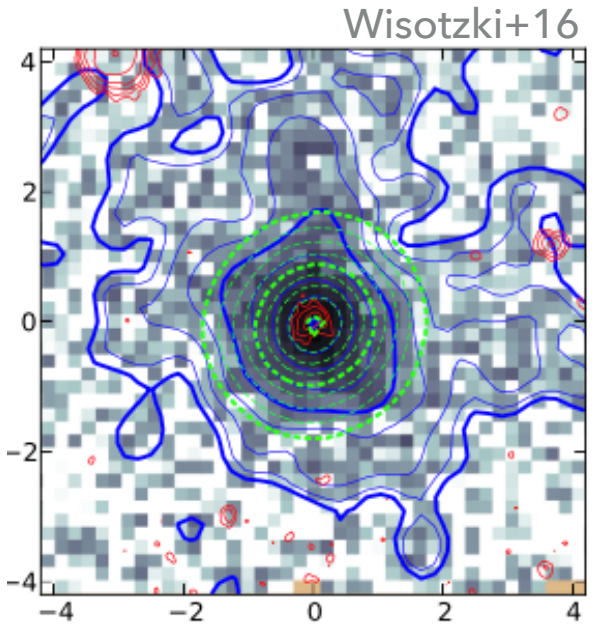
A massively parallel code for line transfer in AMR simulations.

LEO MICHEL-DANSAC, JEREMY BLAIZOT, THIBAULT GAREL, ANNE VERHAMME

SCIENCE MOTIVATIONS

- LAEs, EoR, LABs, … (MUSE, JWST, …)
- CGM in general (Lya, metal line absorption, fluorescence, …)

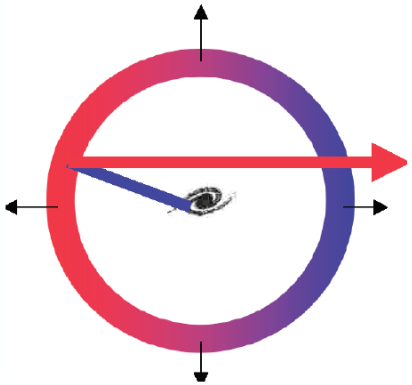


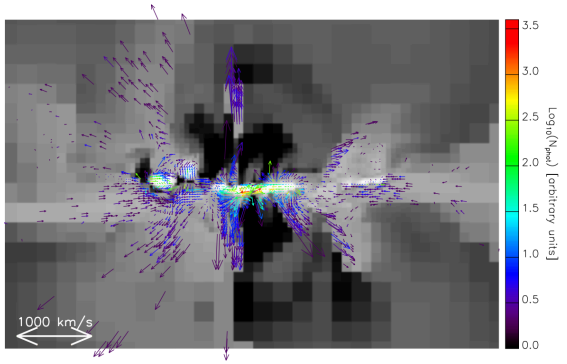


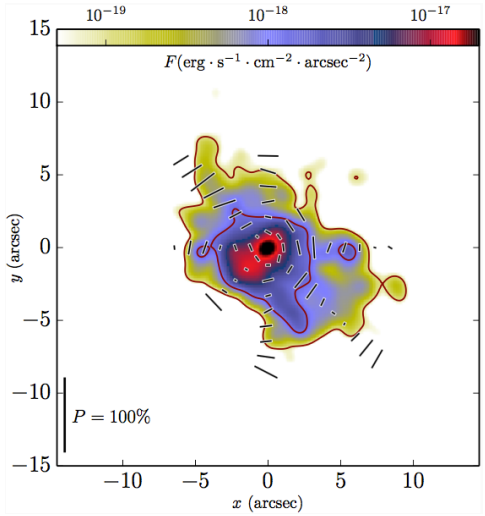
MCLYA (VERHAMME+06,+12, TREBITSCH+16)

A Monte Carlo code for resonant line transfer

- Geneva (Verhamme, Schaerer, Maselli): code development & first application on models of expanding shells (regular mesh)
- Oxford (Verhamme, Dubois): ability to post-process RAMSES simulations (AMR, some level of MPI)
- Lyon (Verhamme, Blaizot, Garel, Trebitsch): optimisation, memory cuts, load balancing (MPI queue), polarisation.







MOTIVATIONS FOR A NEW VERSION

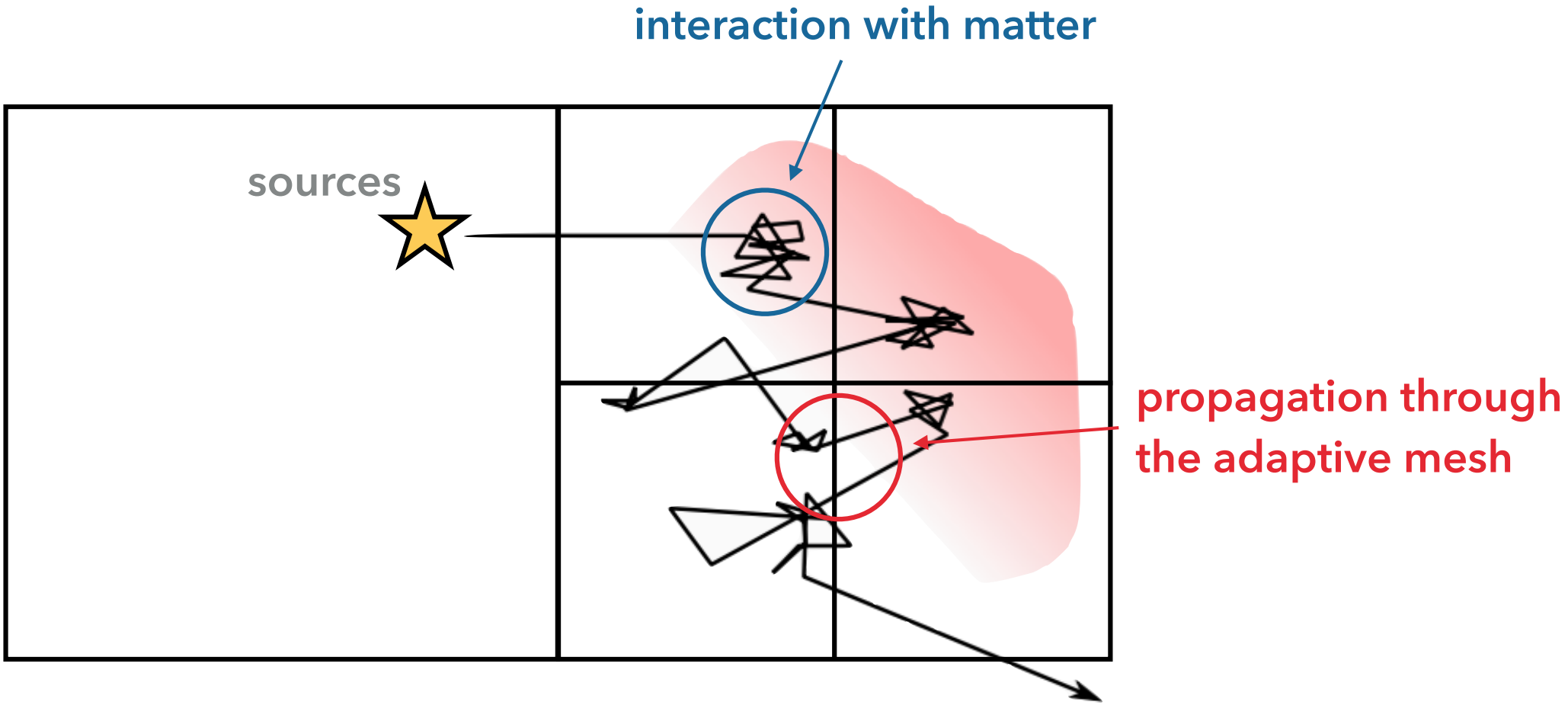
Memory footprint : we need to process large, high-resolution volumes (e.g. haloes of LAEs, giant LABs, HII bubbles at reionisation, ...) on numerous cores (little RAM/core) -> domain decomposition & distributed parallelism.

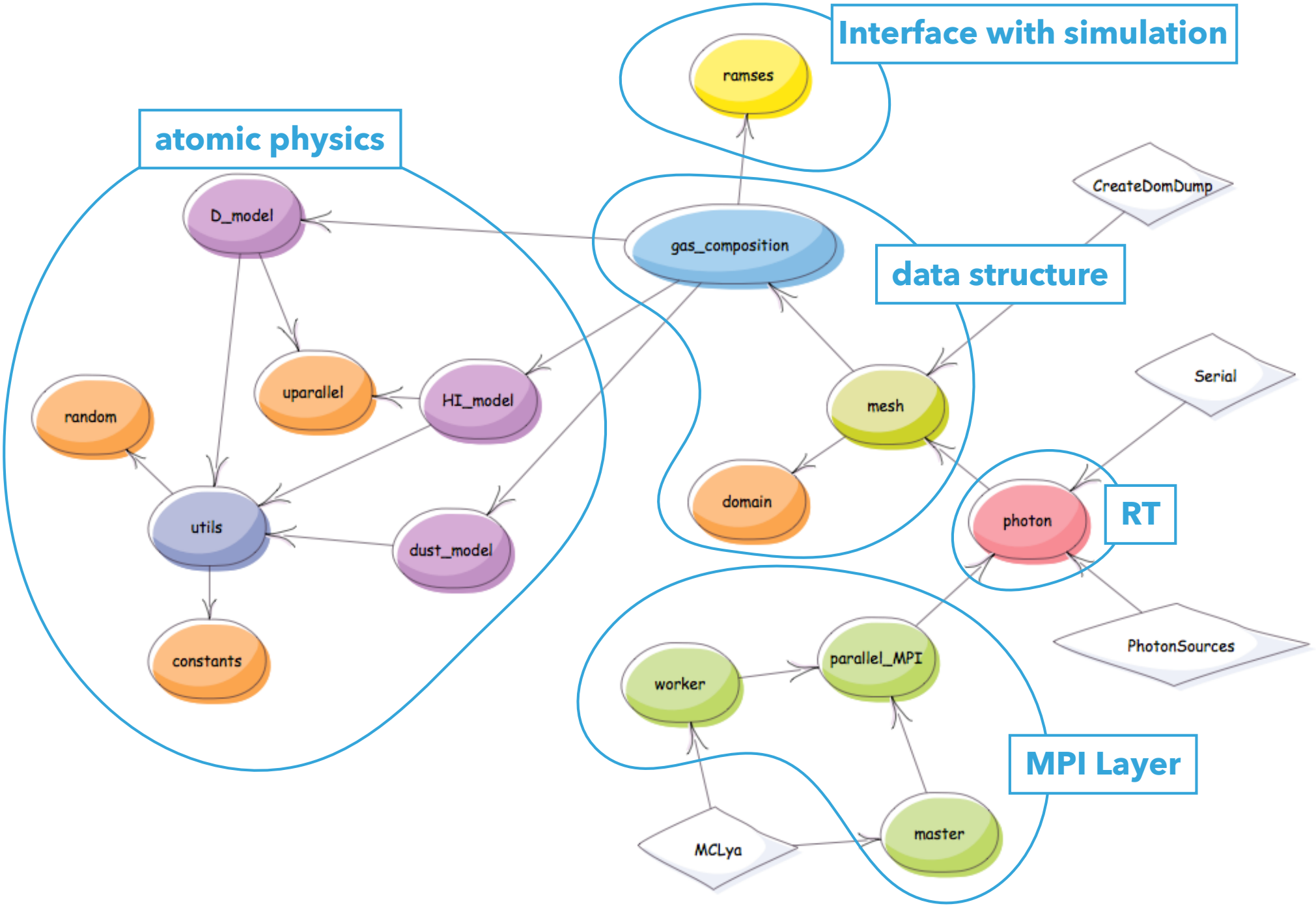
Load balancing: manage the huge diversity of photon effective speeds from low-density IGM to high-density ISM regimes. Balance domain loads.

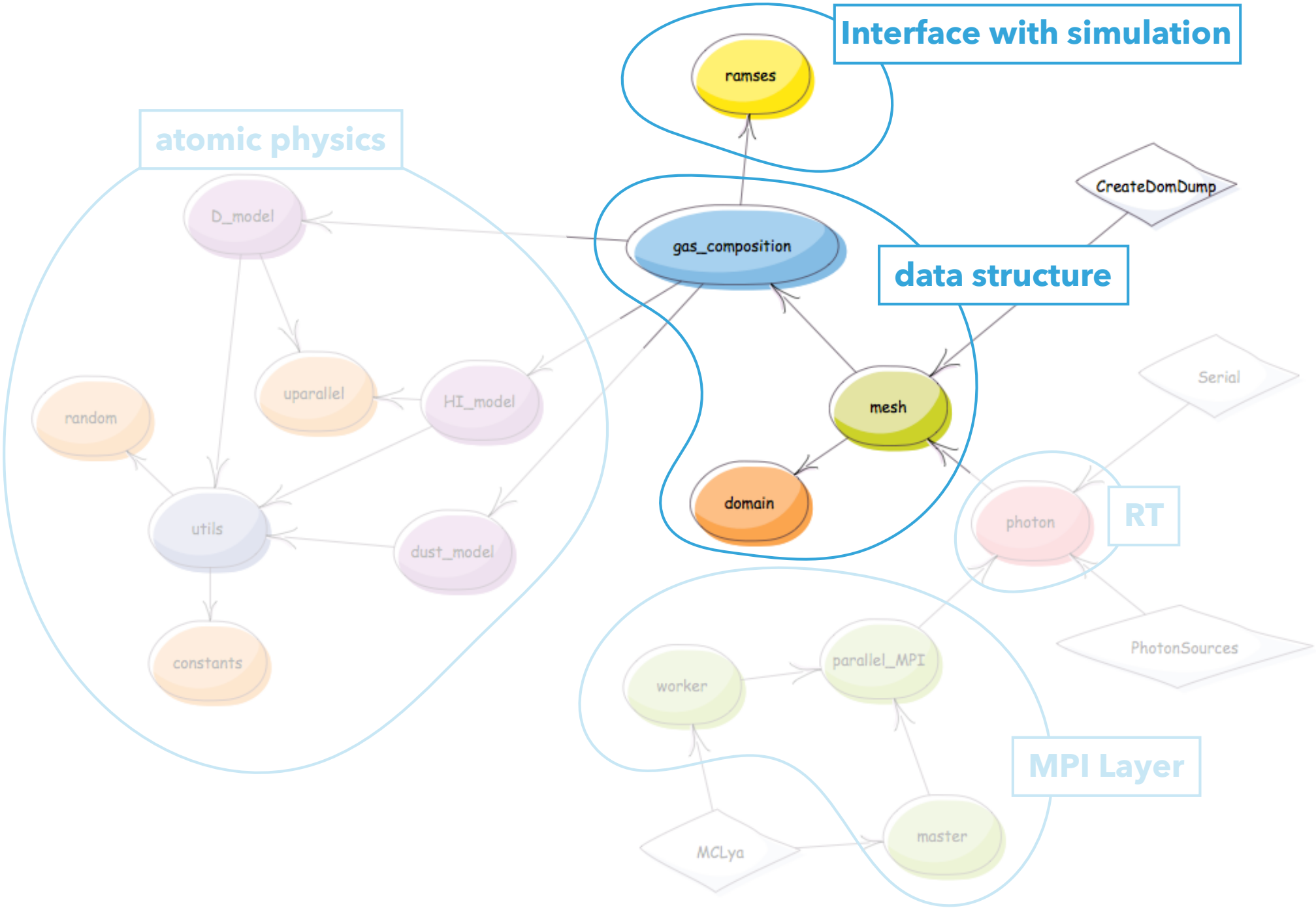
Modularity/community development: Simple add-ons should allow transfer of any line, or processing the outputs of any simulation (other than RAMSES) -> some level of object oriented design / encapsulation.

Some elements of RASCAS should be usable as a toolbox to manipulate RAMSES outputs.

MC LYA RT, TECHNICALLY



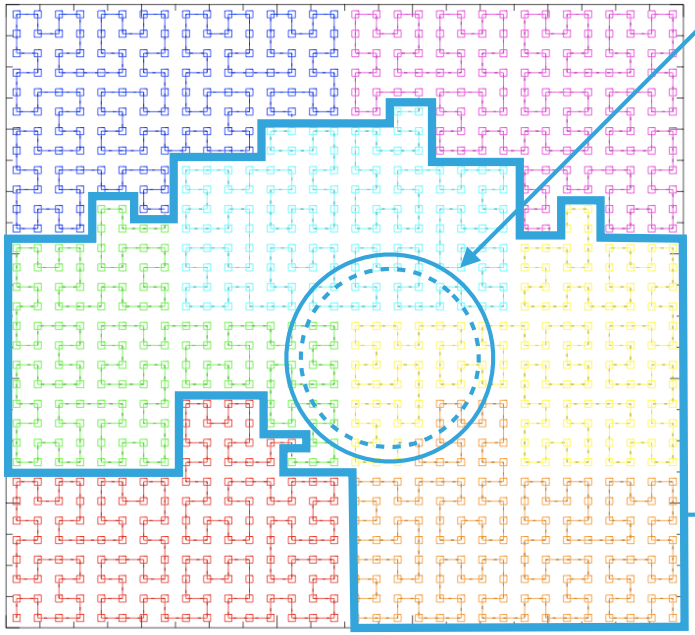

Interface with simulation
ramses
atomic physics
D_model
gas_composition
CreateDomDump
data structure
uparallel
HI_model
random
mesh
Serial
domain
photon
RT
utils
dust_model
constants
PhotonSources
parallel_MPI
worker
MPI Layer
MCLya
master


Interface with simulation
ramses
atomic physics
D_model
gas_composition
CreateDomDump
data structure
uparallel
HI_model
random
mesh
Serial
utils
dust_model
domain
photon
RT
constants
PhotonSources
parallel_MPI
worker
MPI Layer
master
MCLya

DOMAIN (RE)CONSTRUCTION

RAMSES output

Region for RT

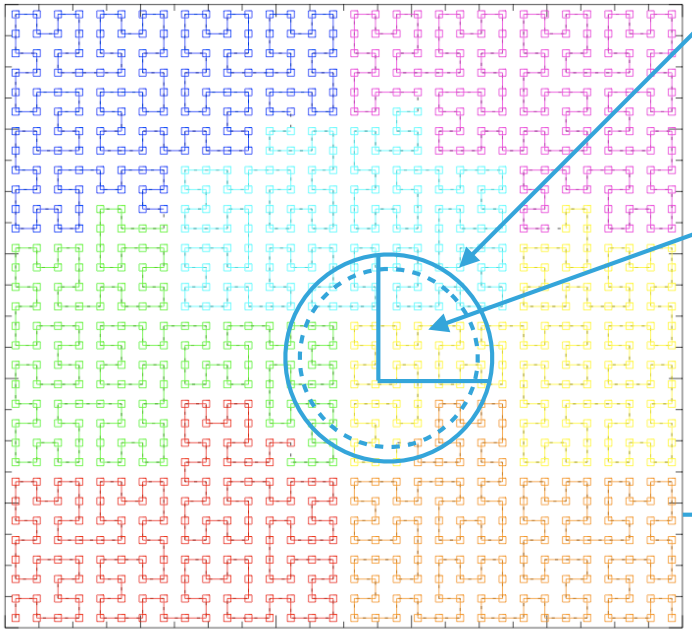


MCLya v1.0 : load all cpu-domains overlapping with zone of interest into memory.

domain-specific indexes
(oct-cell relation, nbor, son, father)

DOMAIN (RE)CONSTRUCTION

RAMSES output



domain-specific indexes
(oct-cell relation, nbor, son, father)

Region for RT

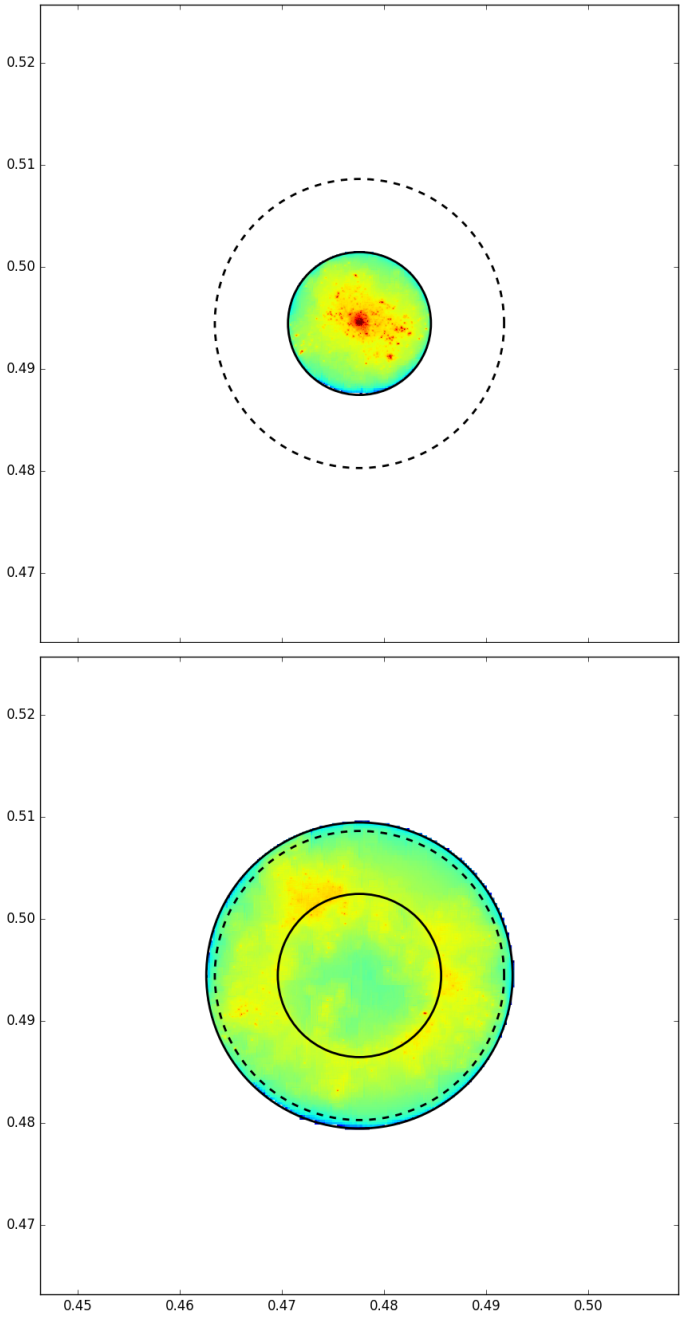
RASCAS: Select an arbitrary collection of leaf cells and re-construct all ramses indexes (oct-tree).
Keep minimal information.

MCLya v1.0 : load all cpu-domains overlapping with zone of interest into memory.

EXAMPLE

Example on a zoom simulation (~14pc resolution, 1E11 Msun halo @ z=3)

- RAMSES output = 18GB
- hydro+AMR data = 9.7GB
- mesh data rebuilt from cells inside Rvir = 1.2GB (4.2GB inside 5*Rvir)
- Can be split further with custom domain decomposition.
- Domains should overlap to avoid photons moving back and forth

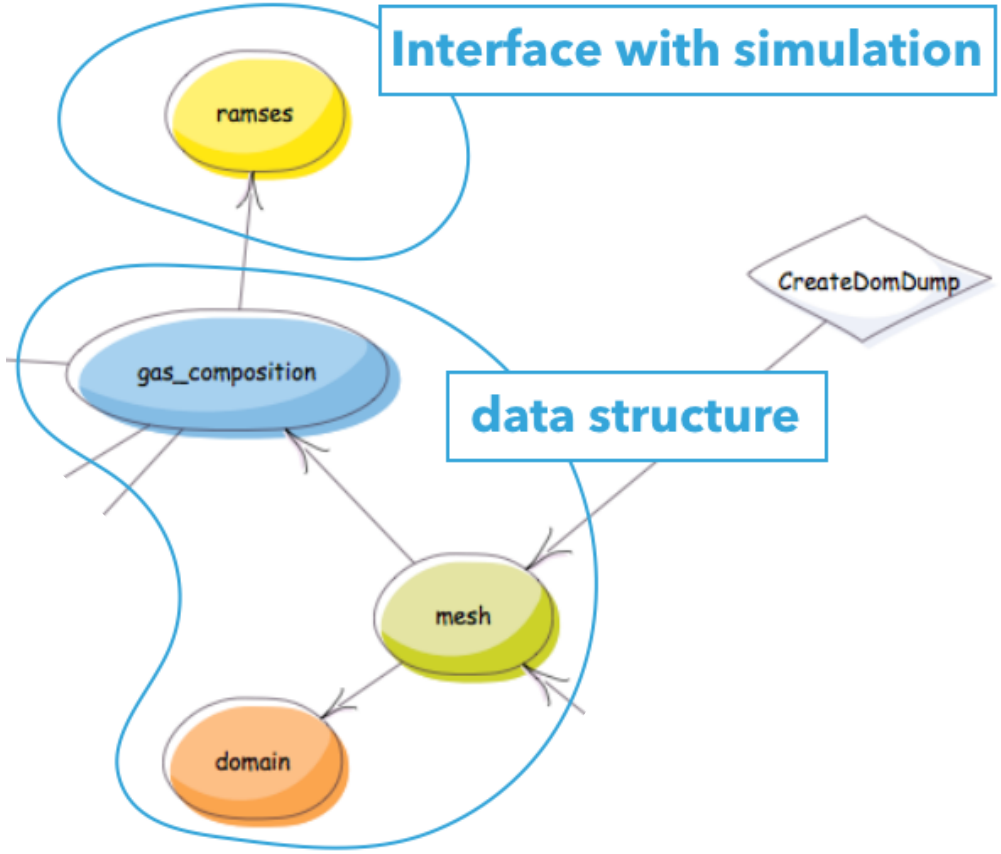


DOMAIN (RE)CONSTRUCTION

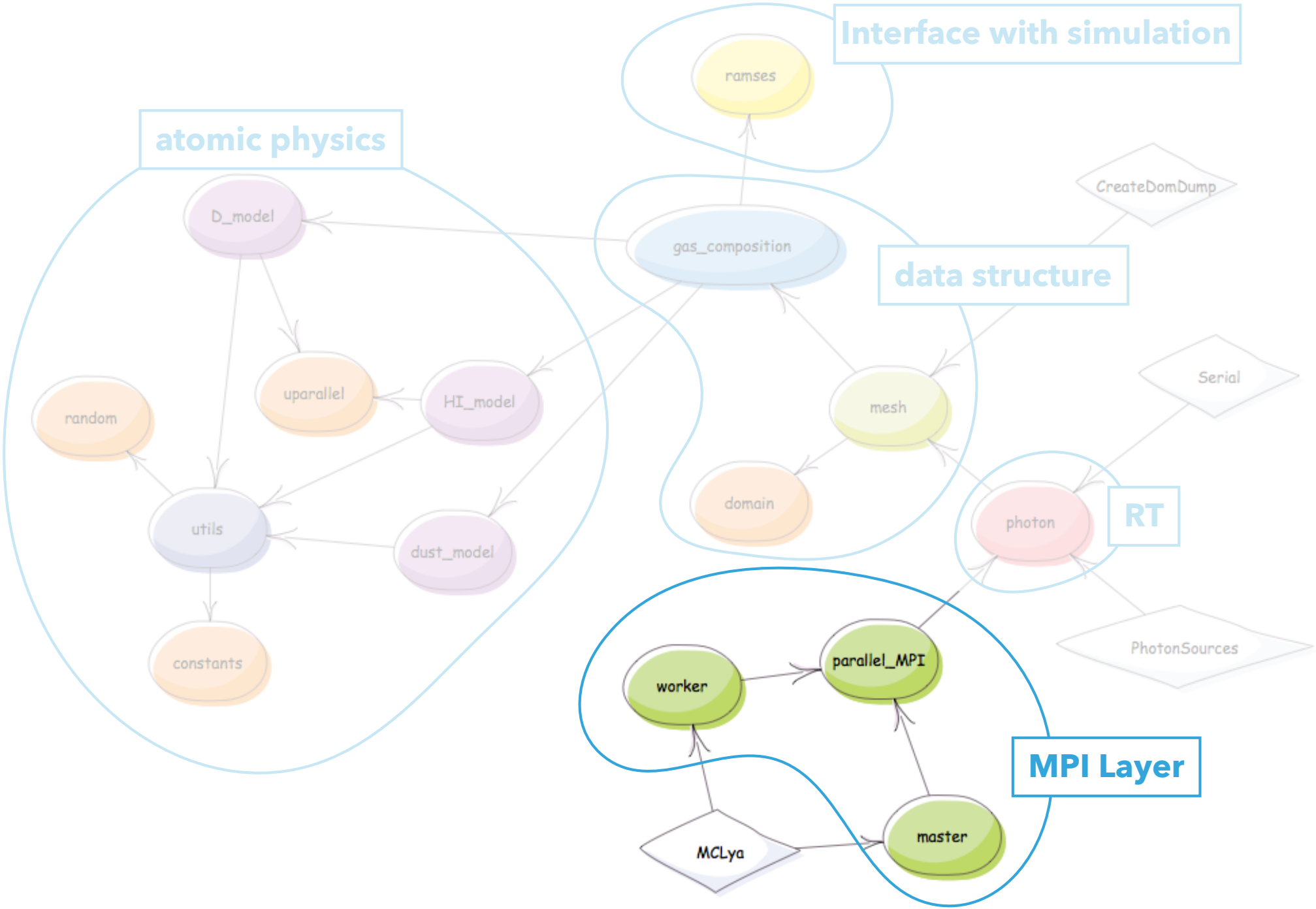
Stand-alone code to build custom domains which contain:

- an AMR mesh with all the oct-tree indexes (son, father, nbor, ...)

- a gas mixture (e.g. HI, Deuterium, and dust) with relevant properties (density, velocity dispersion) for each species.



II. DYNAMICAL LOAD-BALANCING

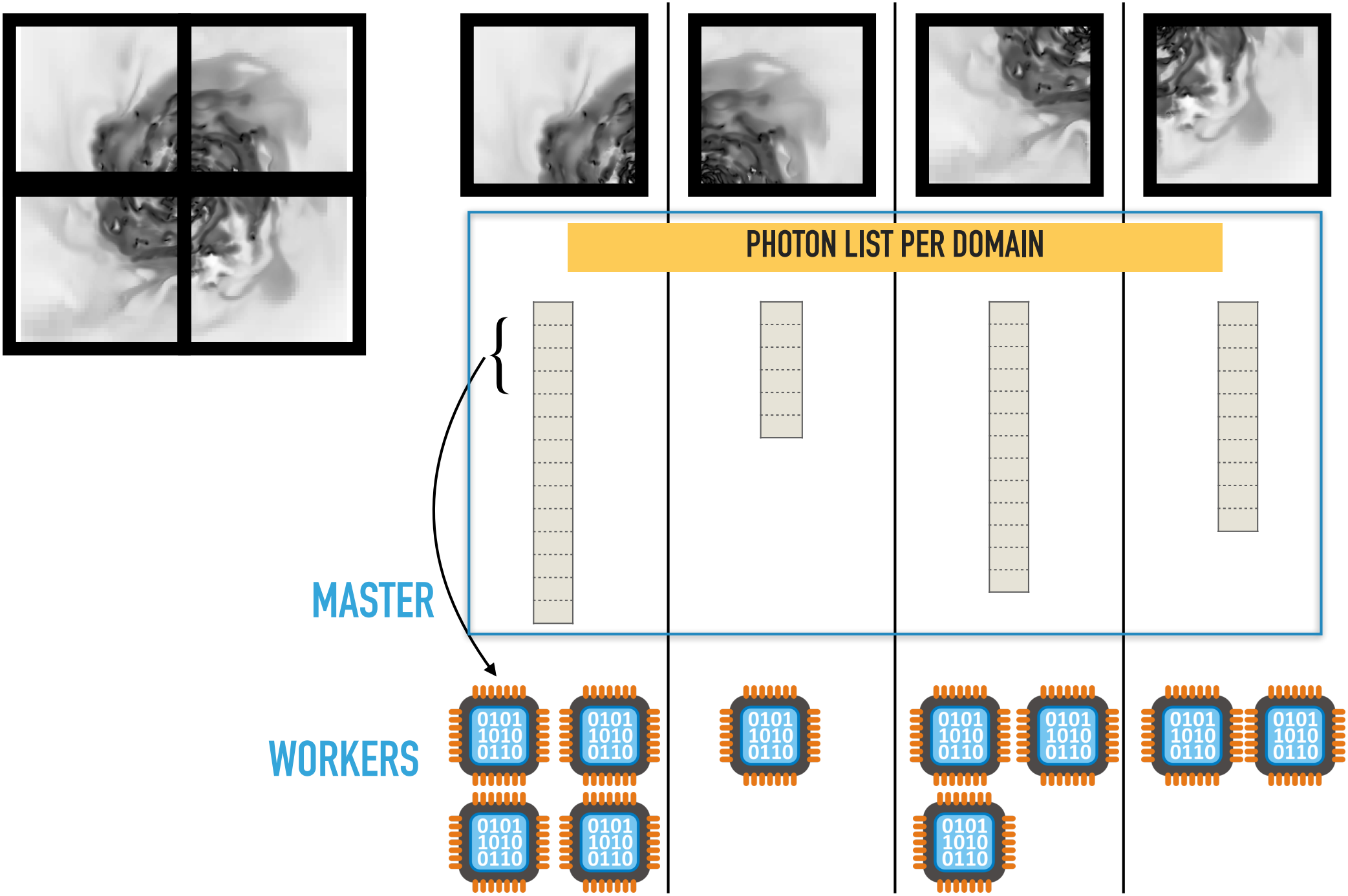


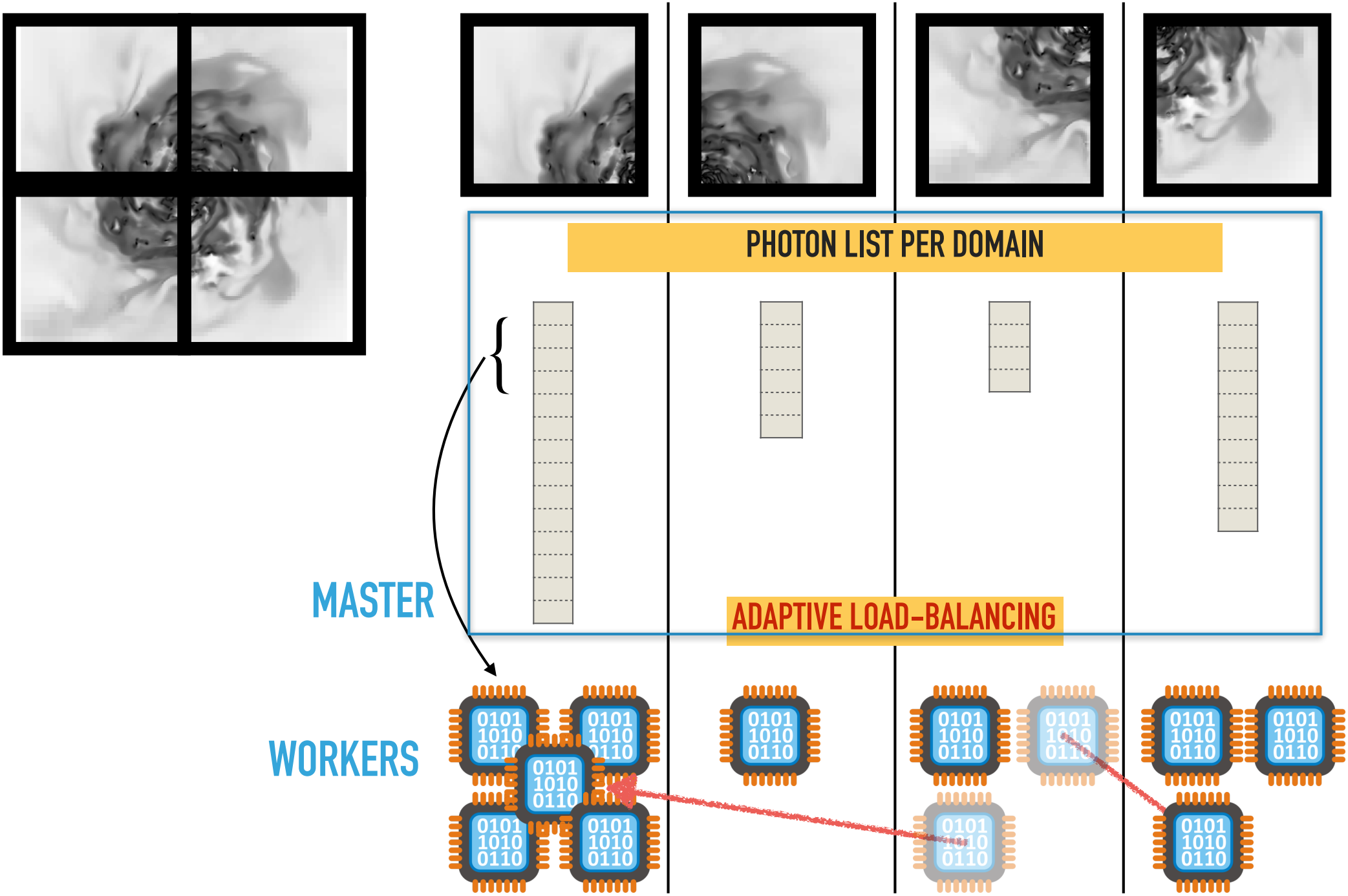
PARALLELISM STRATEGY

Master-slaves scheme, where :

- A slave CPU receives photons from the master, propagates photons through a given domain, and sends them back (when they escape the domain, when the are absorbed, or when the escape the computational volume boundary).

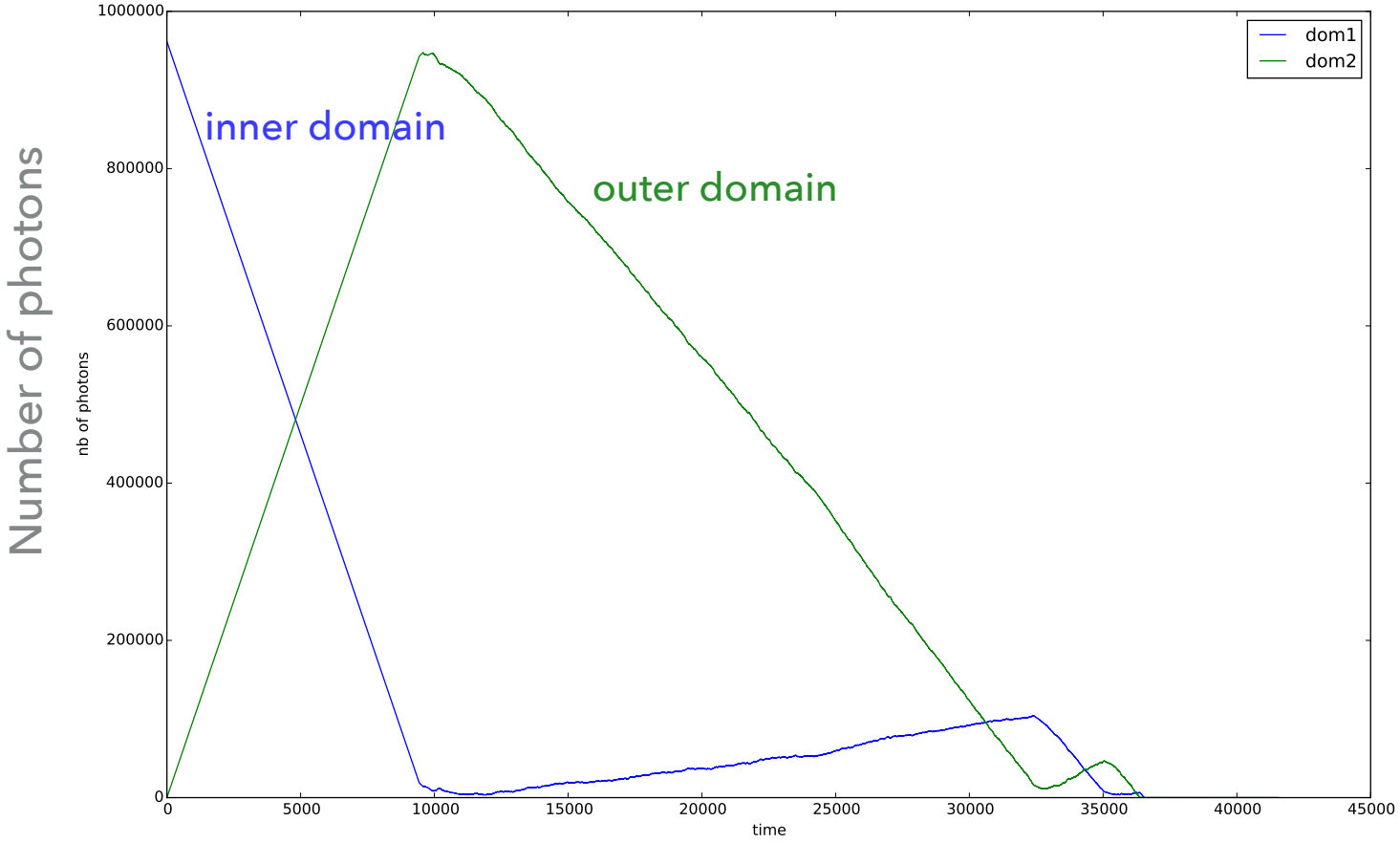
- The master manages **photon queues** (dynamically) and adjusts the ***load per domain*** by assigning more or less CPUs to a given domain.

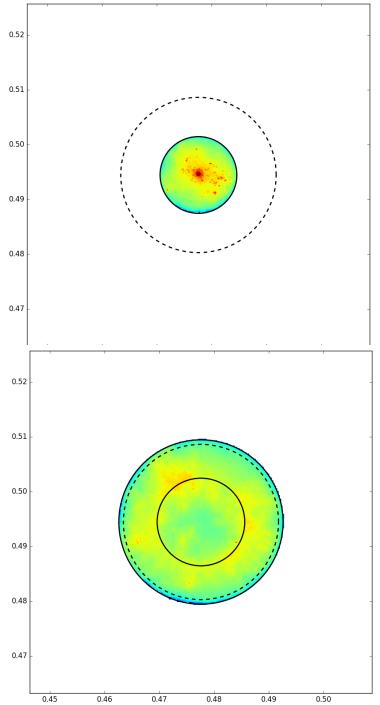

PHOTON LIST PER DOMAIN
MASTER
WORKERS
0101
1010
0110


PHOTON LIST PER DOMAIN
MASTER
ADAPTIVE LOAD-BALANCING
WORKERS
0101
1010
0110

EXAMPLE WITH 2 DOMAINS

CPUs start switching to domain #2



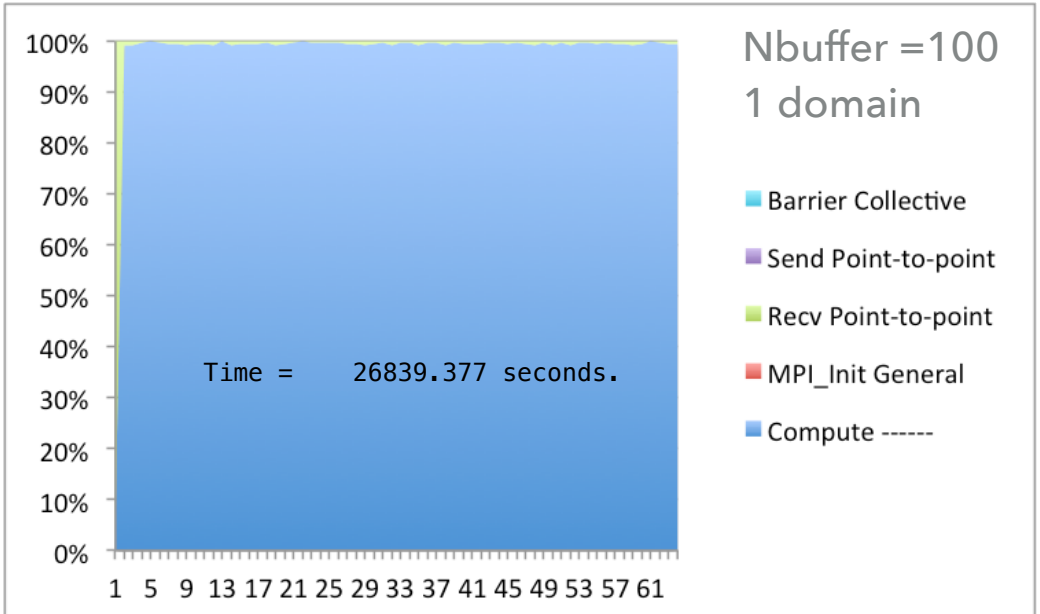


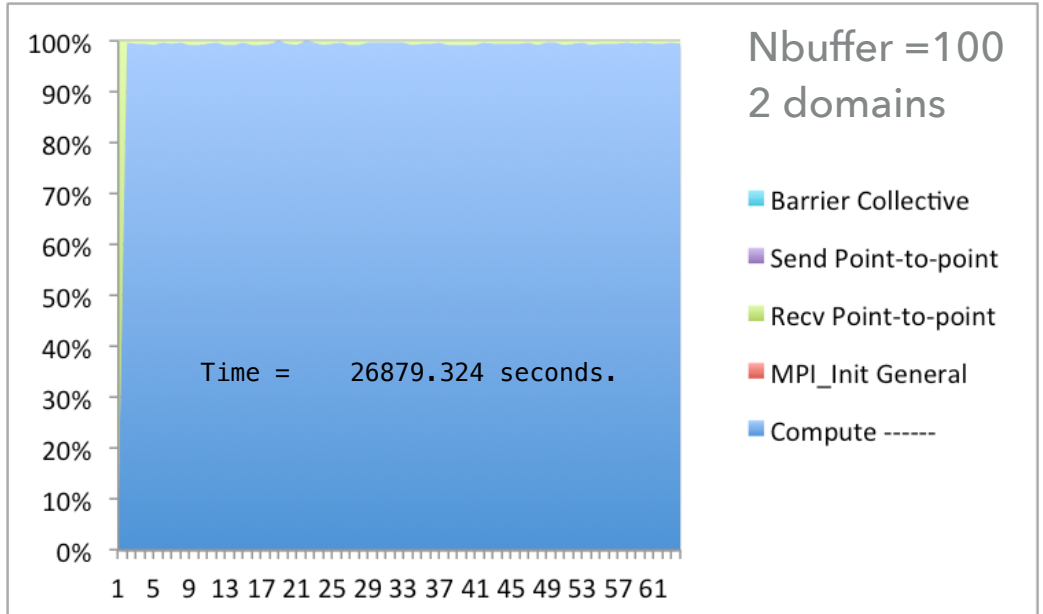
All CPUs on domain #1

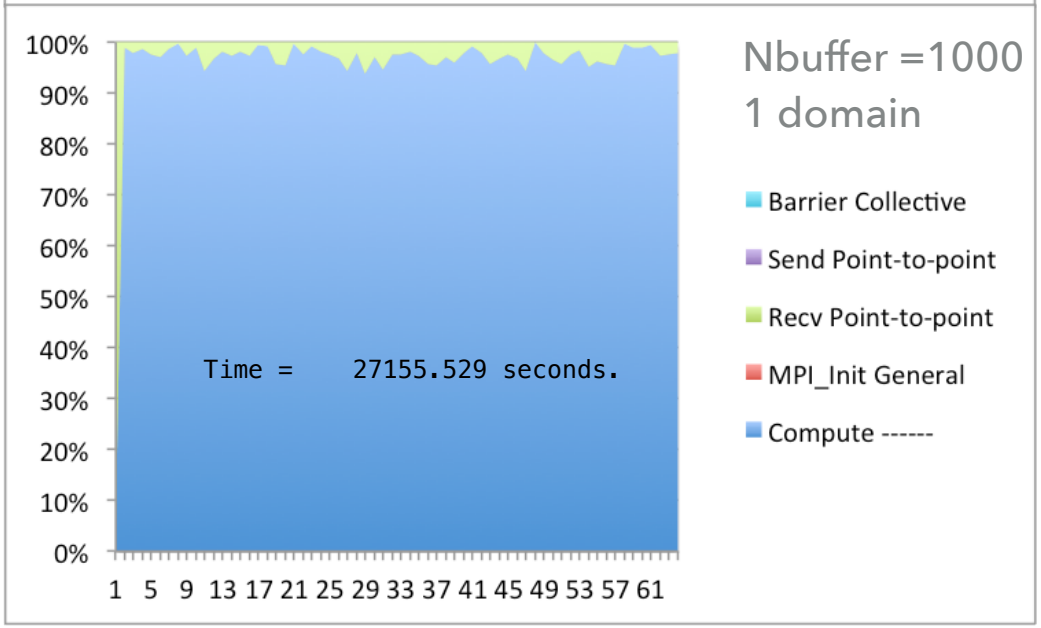
PROFILING MPI

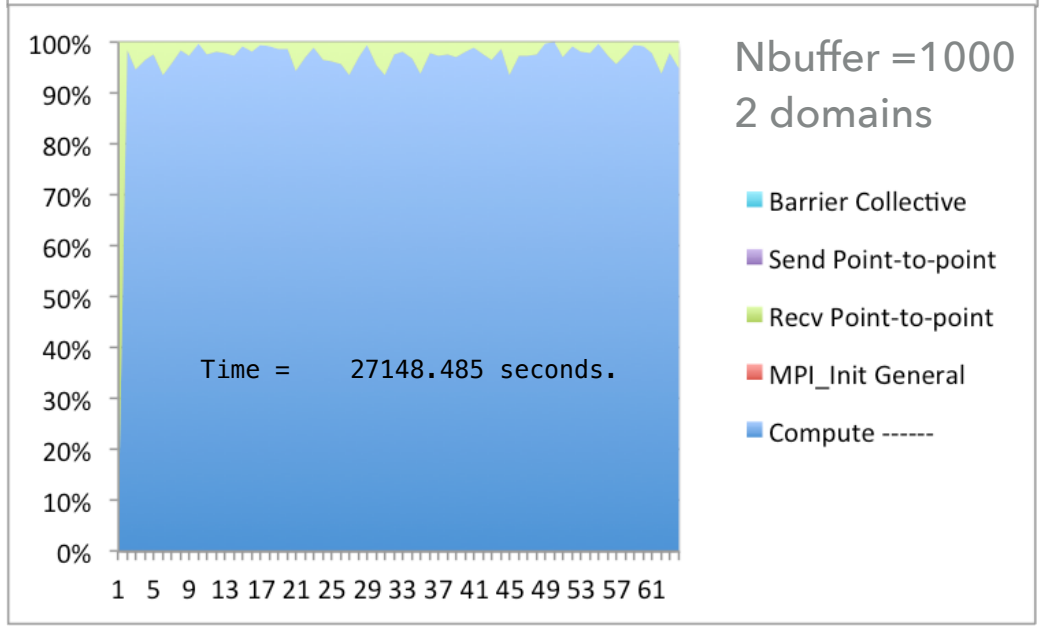
10^6 photons

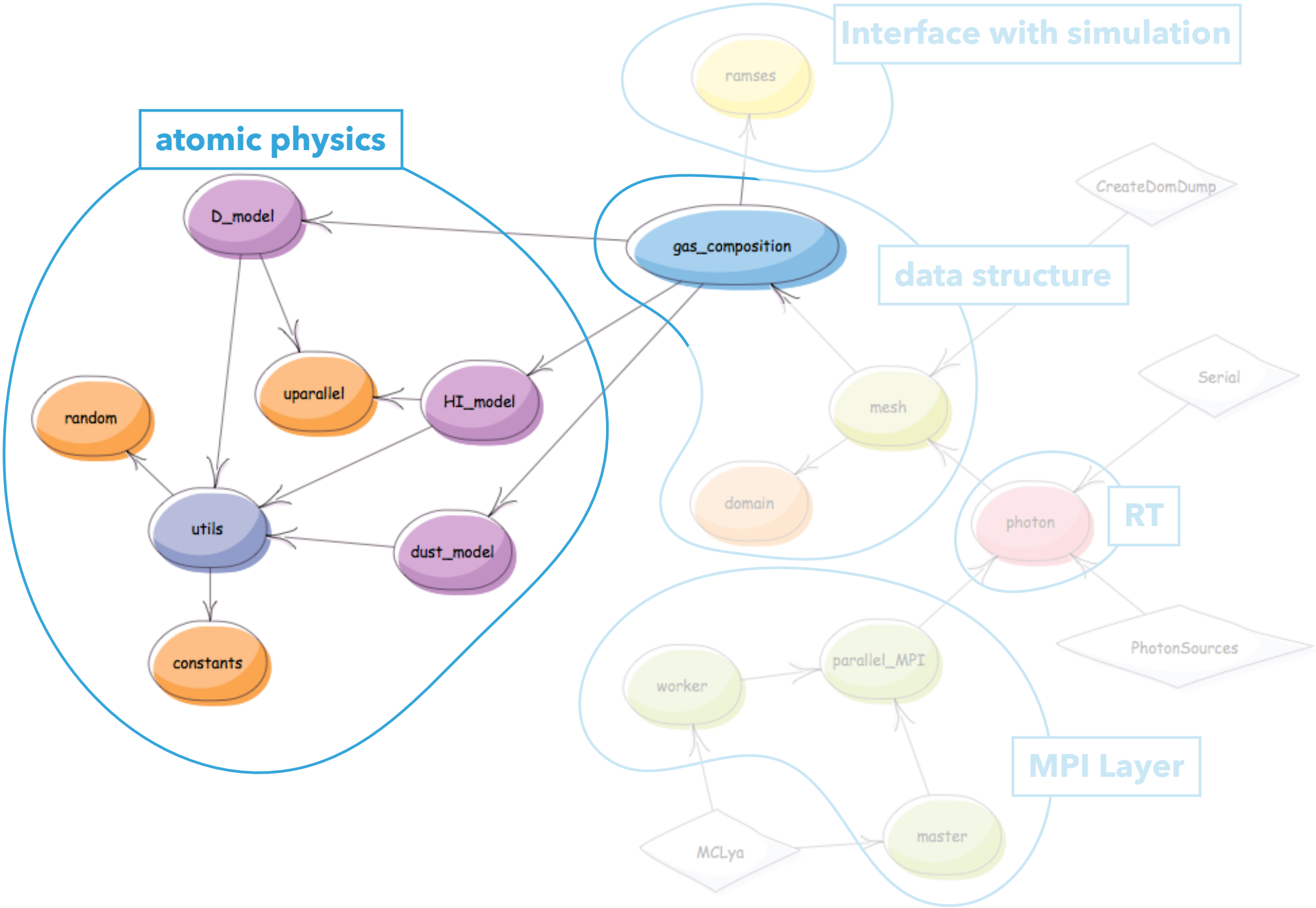
64 threads MPI










Interface with simulation
ramses
atomic physics
D_model
gas_composition
CreateDomDump
data structure
uparallel
HI_model
random
mesh
Serial
domain
photon
RT
utils
dust_model
constants
PhotonSources
parallel_MPI
worker
MPI Layer
master
MCLya

ATOMIC PHYSICS

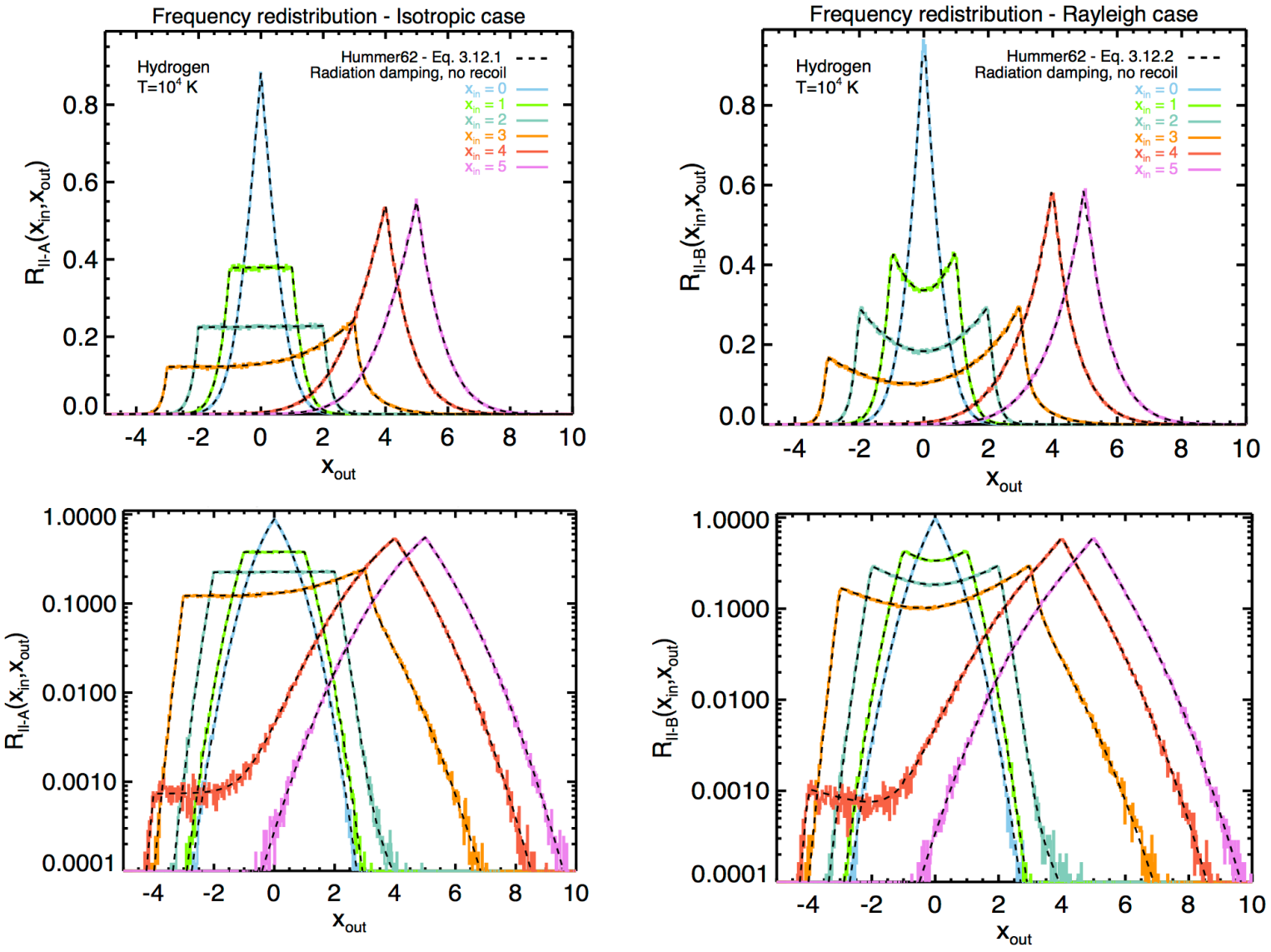
Module “gas composition” asks each scatterer’s module for optical depth and outcome of an interaction.

Scatterer modules (e.g. HI) contain all the atomic physics.

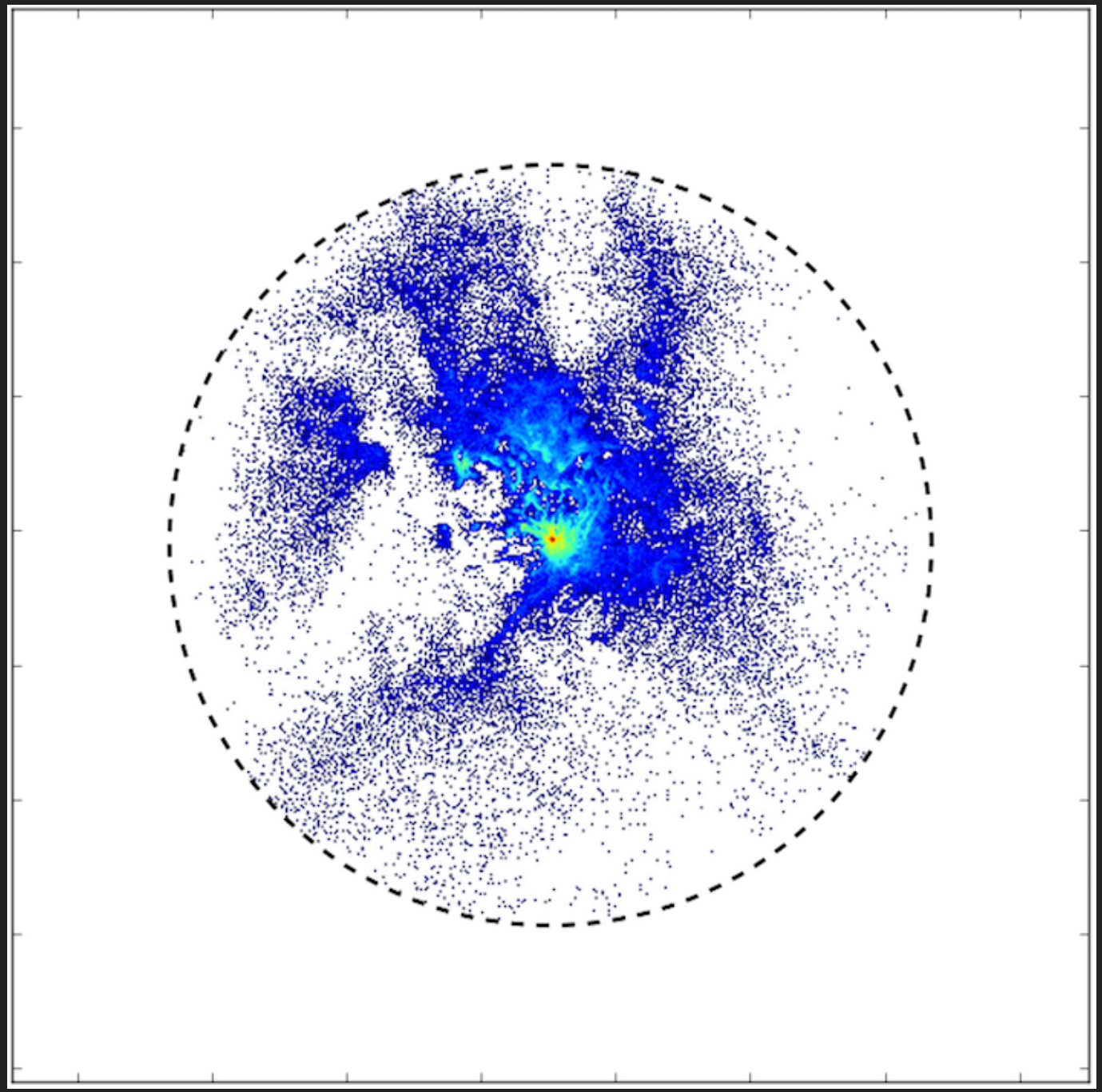
-> easy to add new species/lines (no modification beyond gas composition module).

-> allows to robust testing of each element.

REDISTRIBUTION FUNCTIONS



III. RADIATIVE TRANSFER - EXAMPLE RUN ON A ZOOM SIMULATION



FUTURE DIRECTIONS

- Use for science ! (LAEs, LABs, EoR, …)
- Add more physics (more lines, e.g. SiII, MgII, etc.) to probe CGM.
- Polish some functionalities (restart, random number, …) and explore MPI parameters to get optimal perf.
- Implement some optimisations for Lya.
- Develop interface(s) with other codes ?
- Try GPUs ?

THANK YOU

LEO MICHEL-DANSAC, JEREMY BLAIZOT, THIBAULT GAREL, ANNE VERHAMME