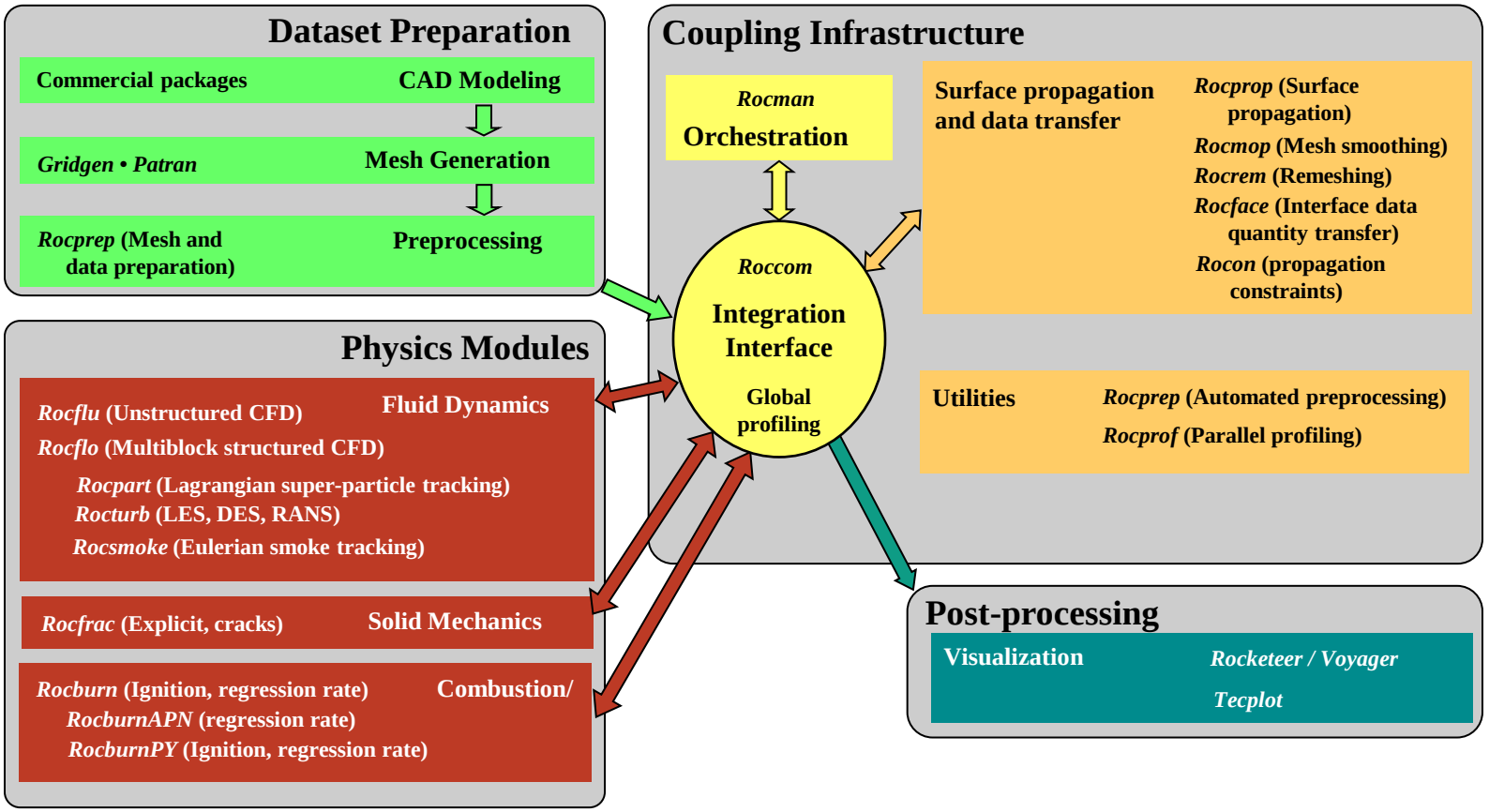
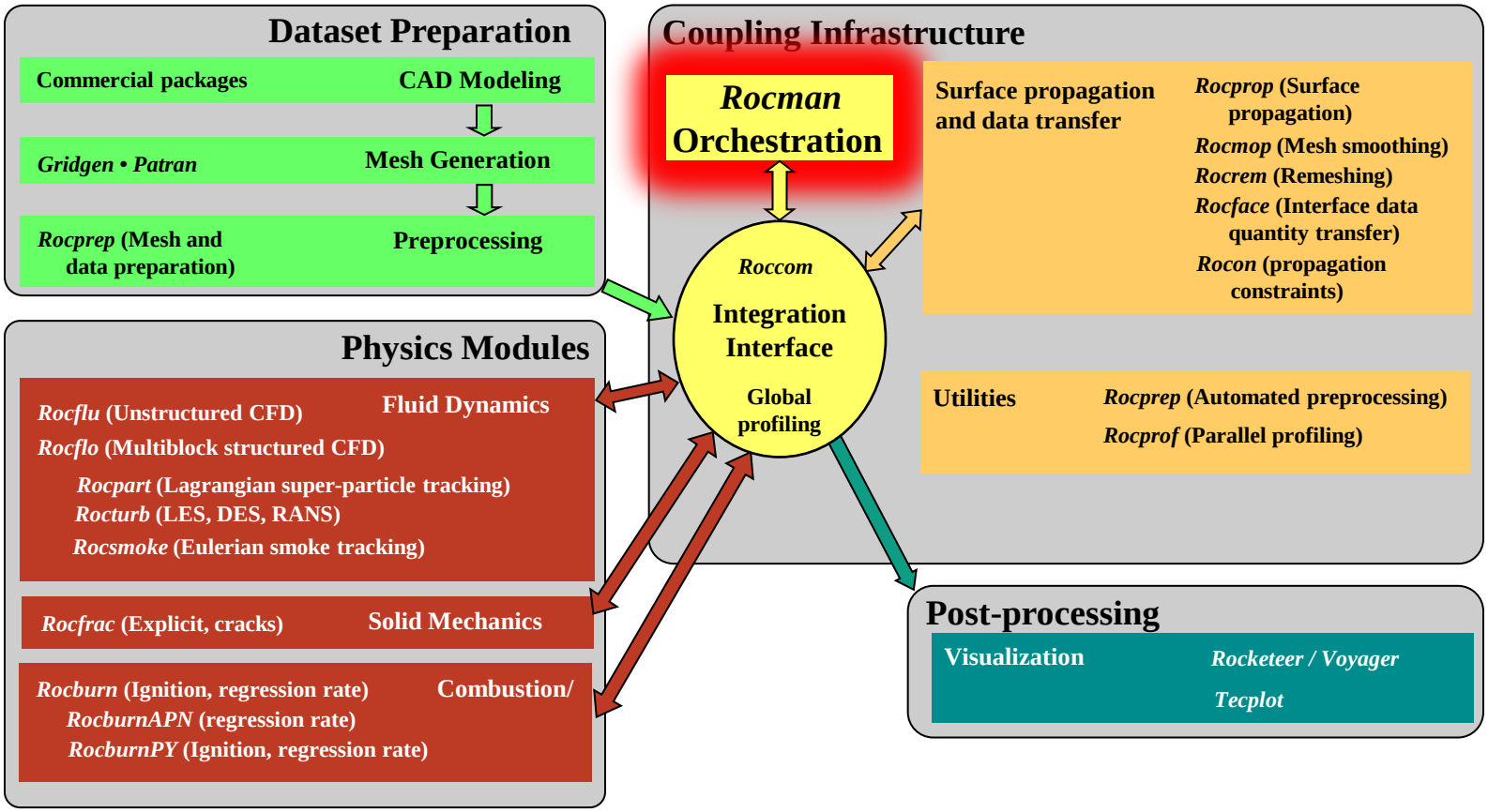
Section 5
Rocman Orchestration

Rocstar Simulation Suite Architecture



Rocstar Simulation Suite Architecture

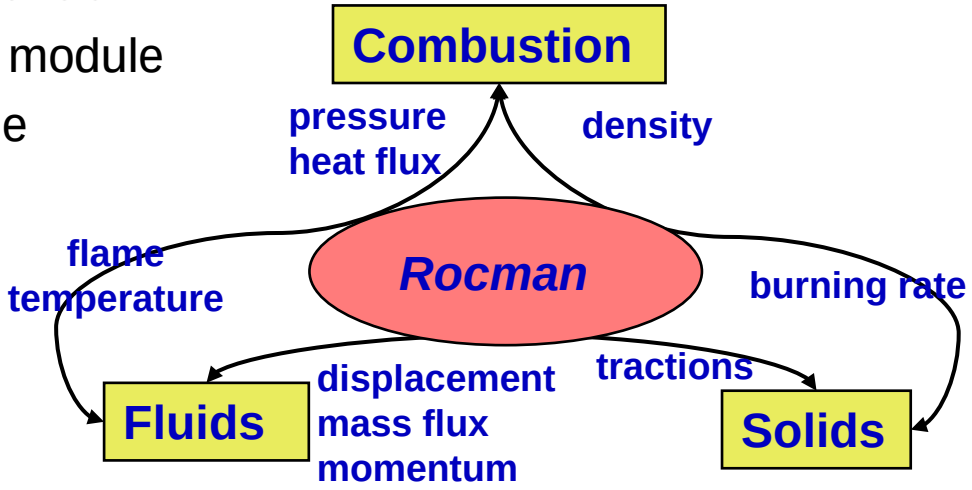


Component Management and Orchestration: *Rocman*

- Manages control flow
- Manages global parallelism using MPI
- Simulation Orchestration API
  - Implements time stepping schemes
  - Provides sub-stepping callbacks
  - Handles jump conditions
  - Performs unit conversions
- Module/process mapping
- Generic control callbacks
  - Interrupts

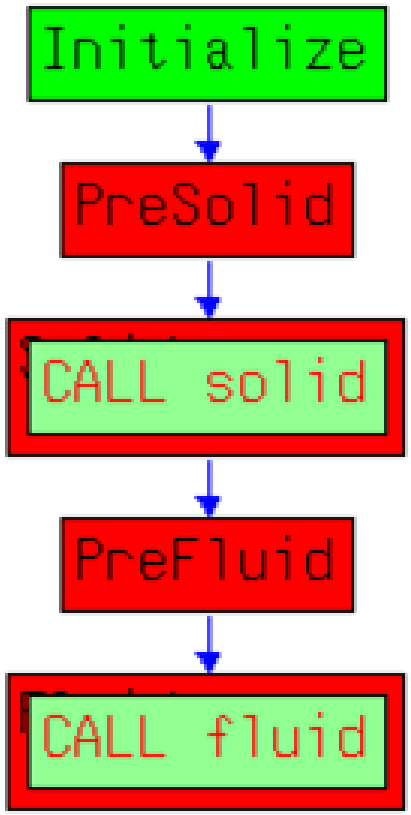
Orchestration Framework (*Rocman*)

- Coupled simulation involves interactions of multiple physics and service modules
  - Requires a high level control module
  - Physics modules must provide
    - Initialization
    - Update
    - Compute integrals
    - Finalize



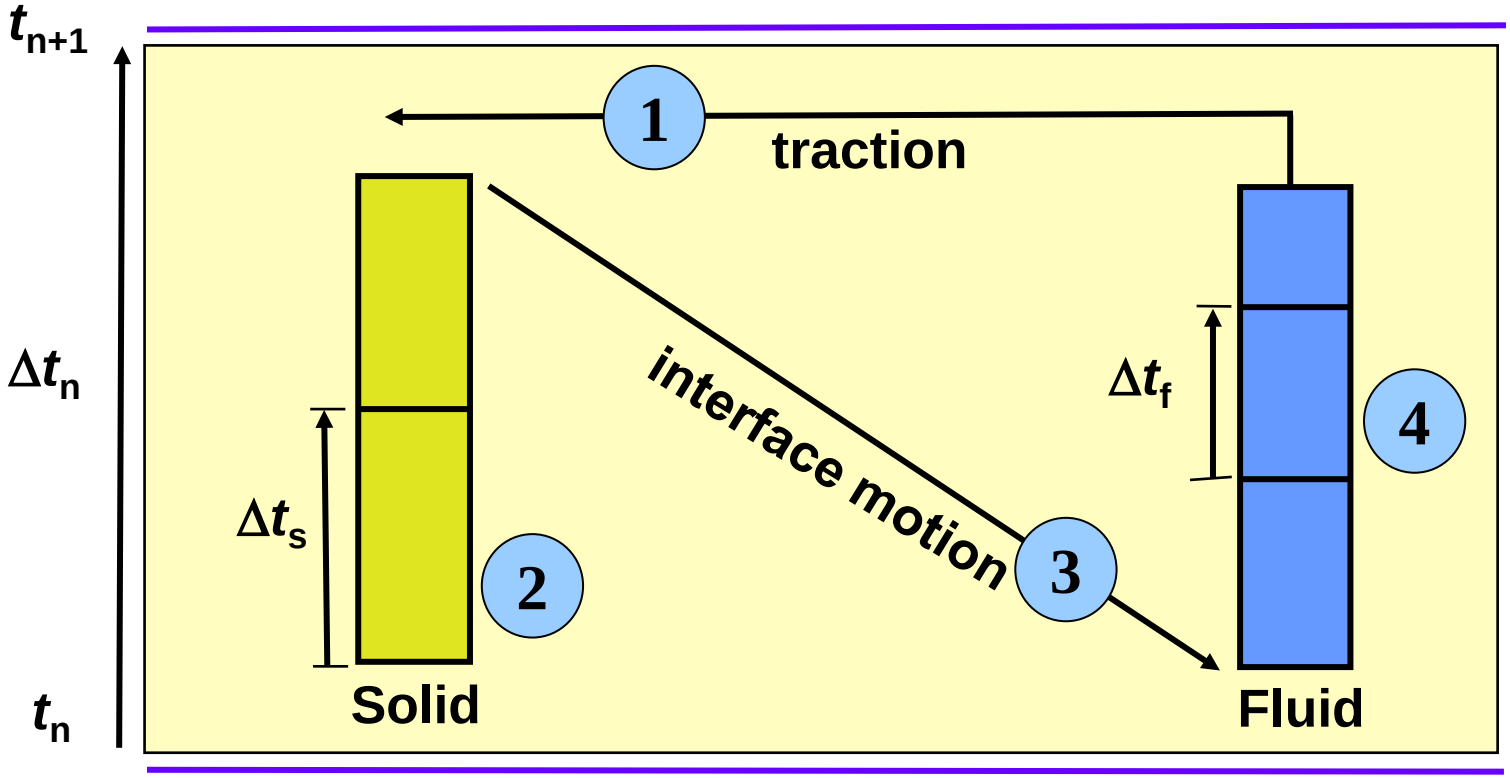
- Task management
  - Specify control flow with calling mechanism of *Roccom*
  - Manipulate jump conditions using service utilities
- Data management
  - Allocate intermediate buffers for communication
  - Initiates/manages I/O for restart and visualization

Hierarchical Architecture



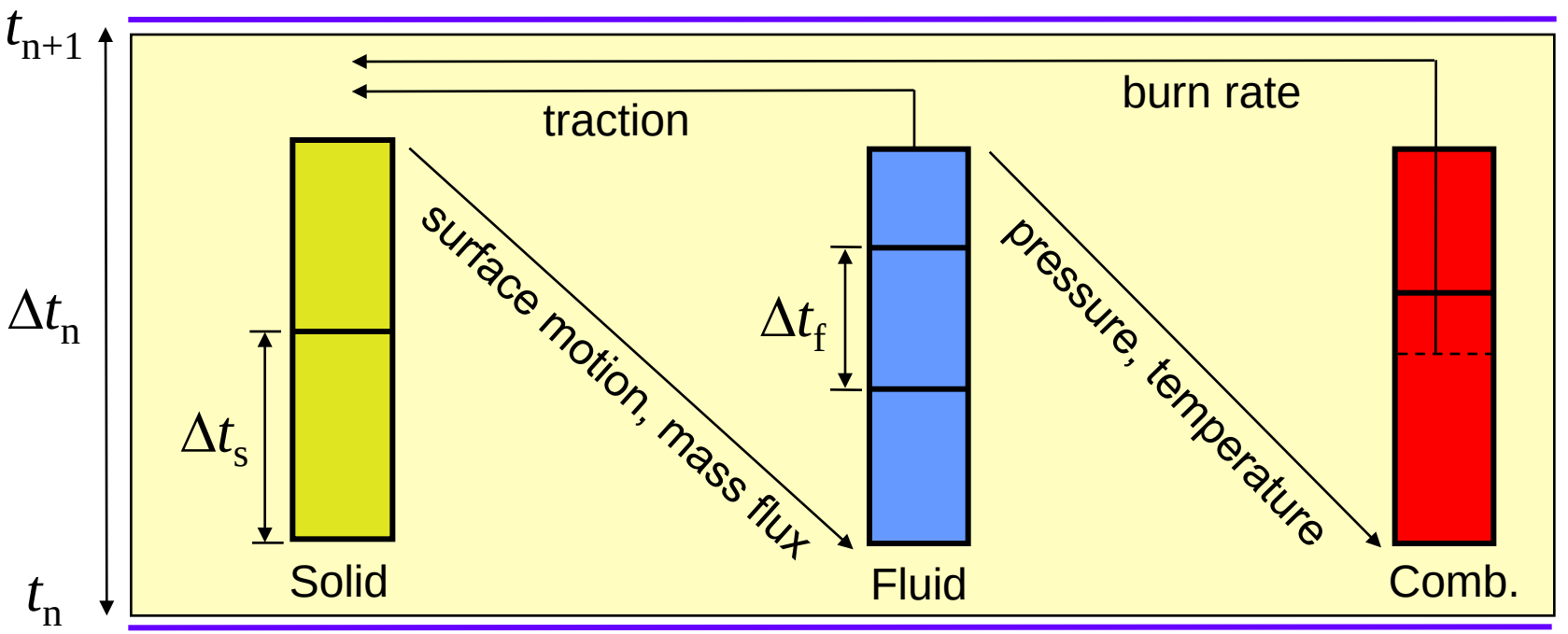
- Top-level iterations and coupling modes
  - Time-marching schemes for both steady and unsteady-state calculations
- Agents for physics modules
  - Represents a domain-specific simulation (fluid, solid or combustion)
- Action
  - A functional object implementing a designated calculation
  - Self-contained with input and output specification
- Schedulers
  - Containers of actions
- Coupling objects
  - Composed of a number of agents and a scheduler
  - Only code a user needs to write

Example – Solid/Fluid Coupling



① Tractions transferred from fluid interface mesh to solid
② Solid physics module computes displacement of the interface
③ Transfer the (interpolated) displacement of the interface to fluid
④ Fluid module solves for tractions

Basic Time Stepping Schemes



- Execution: 1) all solid, 2) all fluid, 3) all combustion
- Updated interface data is passed through Rocface
- Components can adaptively subcycle
- Predictor/corrector: repeat step until solution converges