Multiscale Materials Design: An environment for multiscale simulations

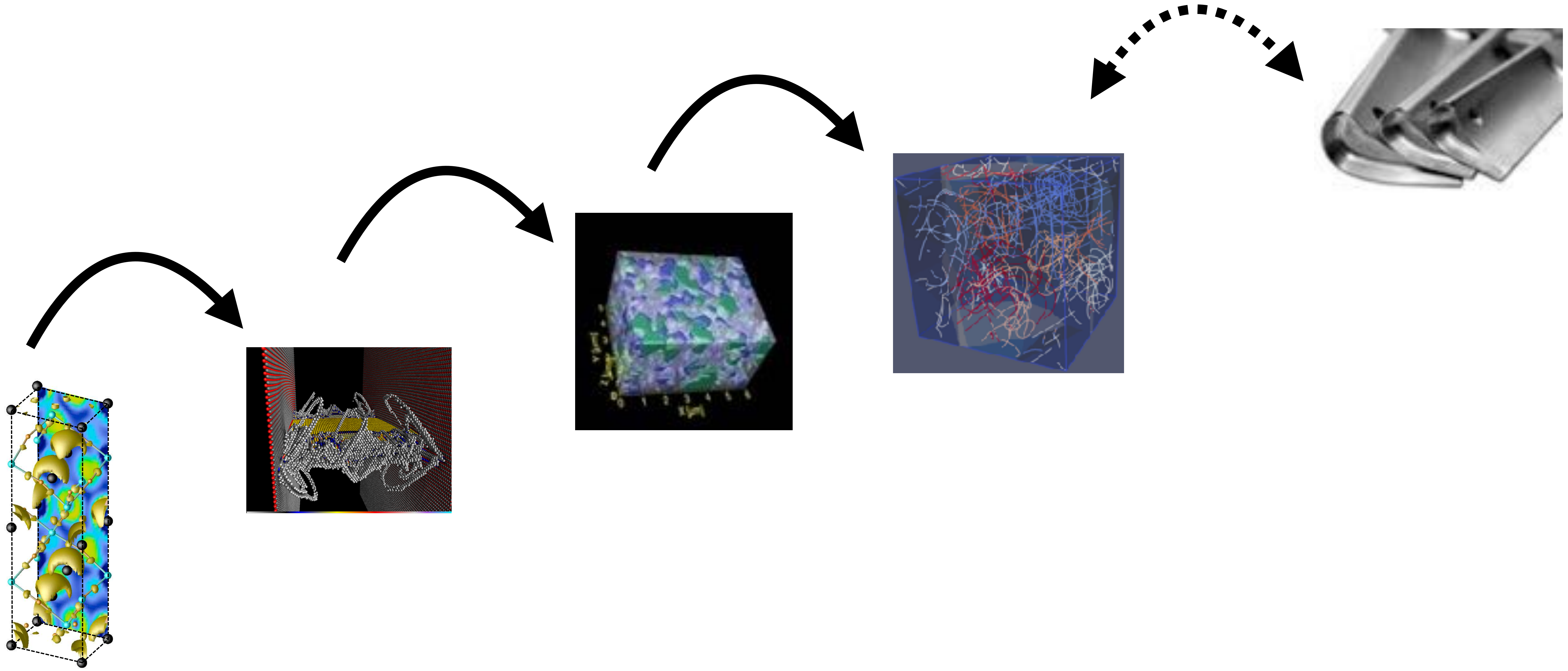
Richard LeSar
Kenneth M. Bryden

Graduate students:

- Laurel Barnet
- Tina Akinyi

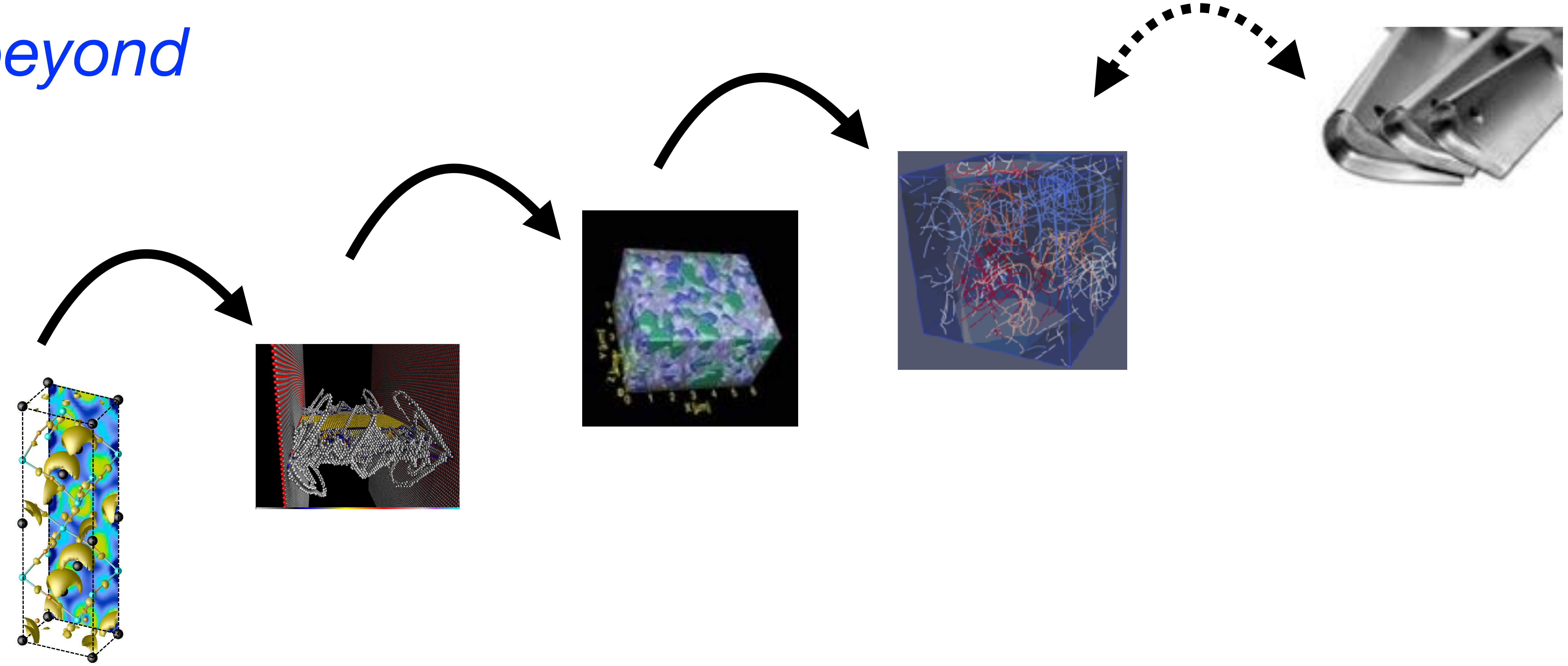
Team

Integrated Computational
Materials Engineering (ICME)



Integrated Computational
Materials Engineering (ICME)

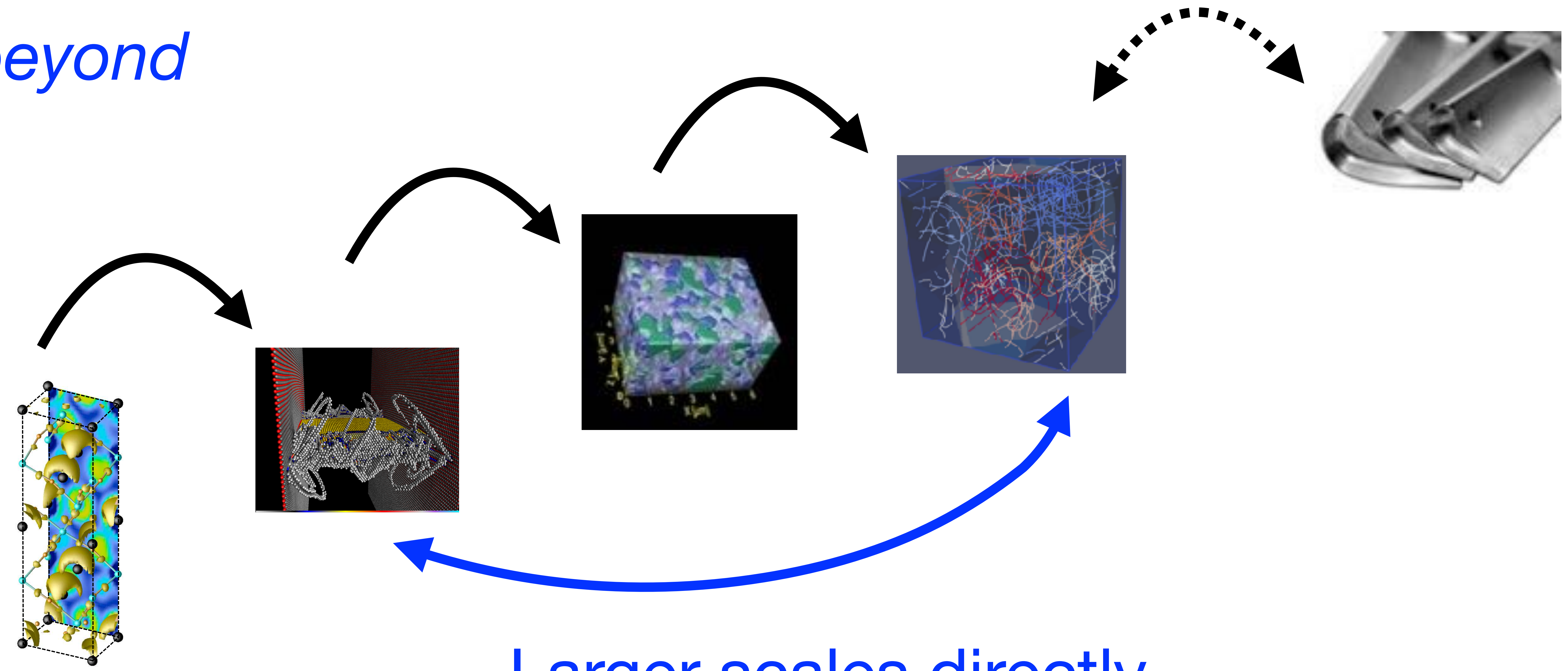
Goal is to link materials into product design — beyond ICME



Materials design — beyond ICME

Integrated Computational Materials Engineering (ICME)

Goal is to link materials into product design — beyond ICME

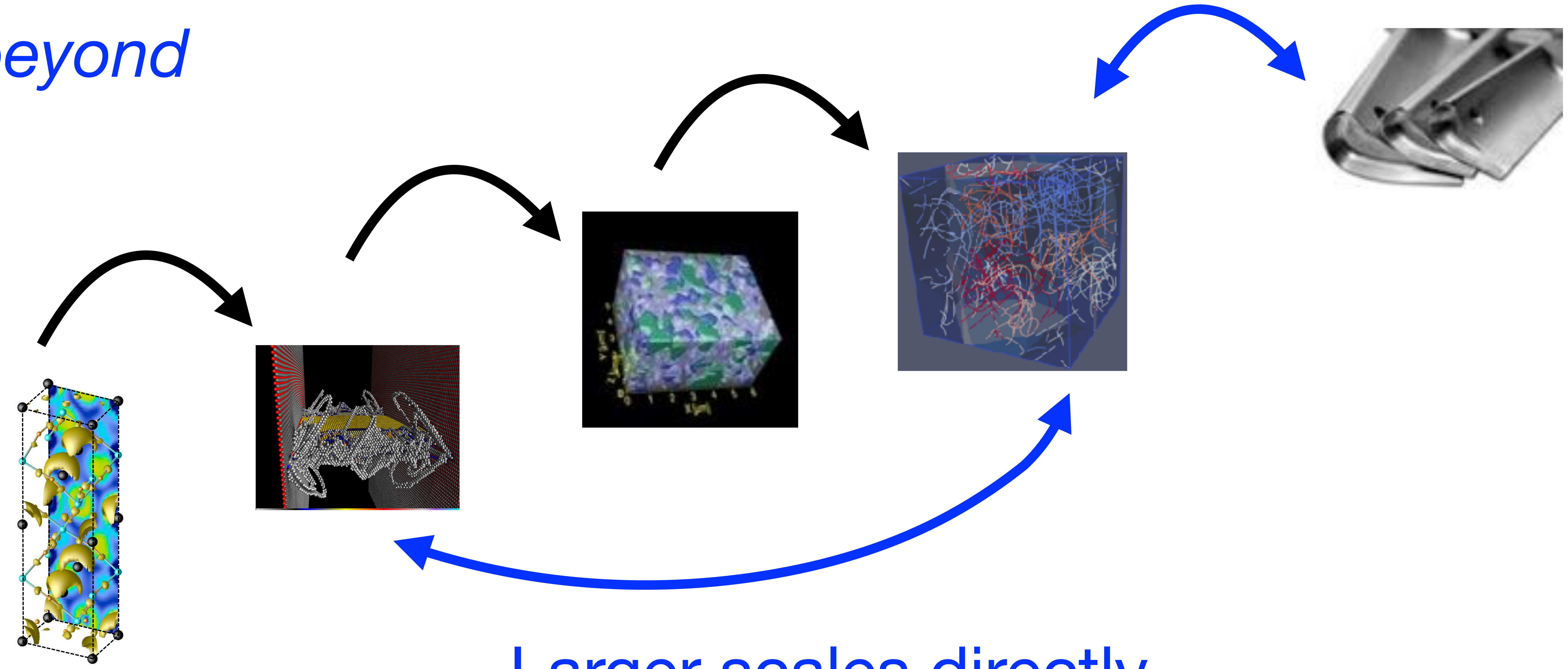


Larger scales directly inform smaller scales

Integrated Computational Materials Engineering (ICME)

Goal is to link materials into product design — beyond ICME

Feedback from part to material

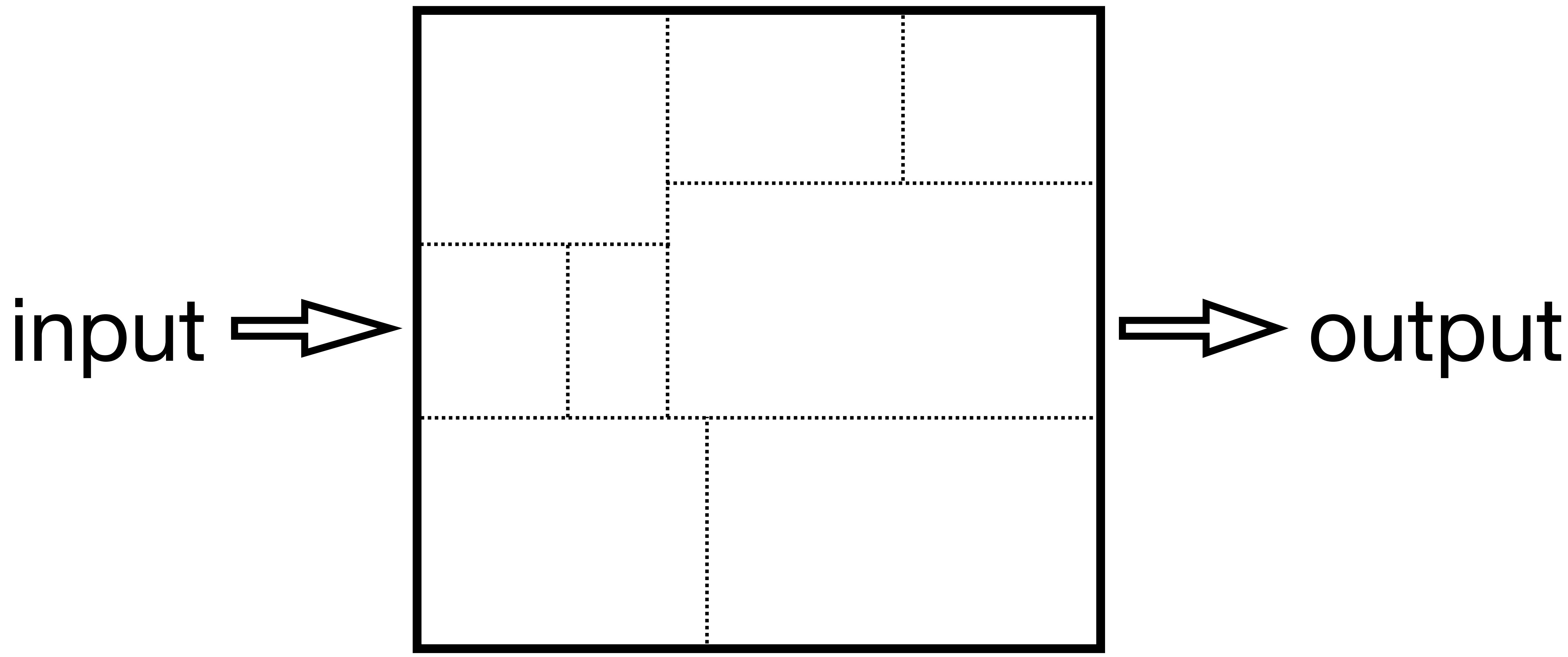


Larger scales directly inform smaller scales

Materials design — beyond ICME

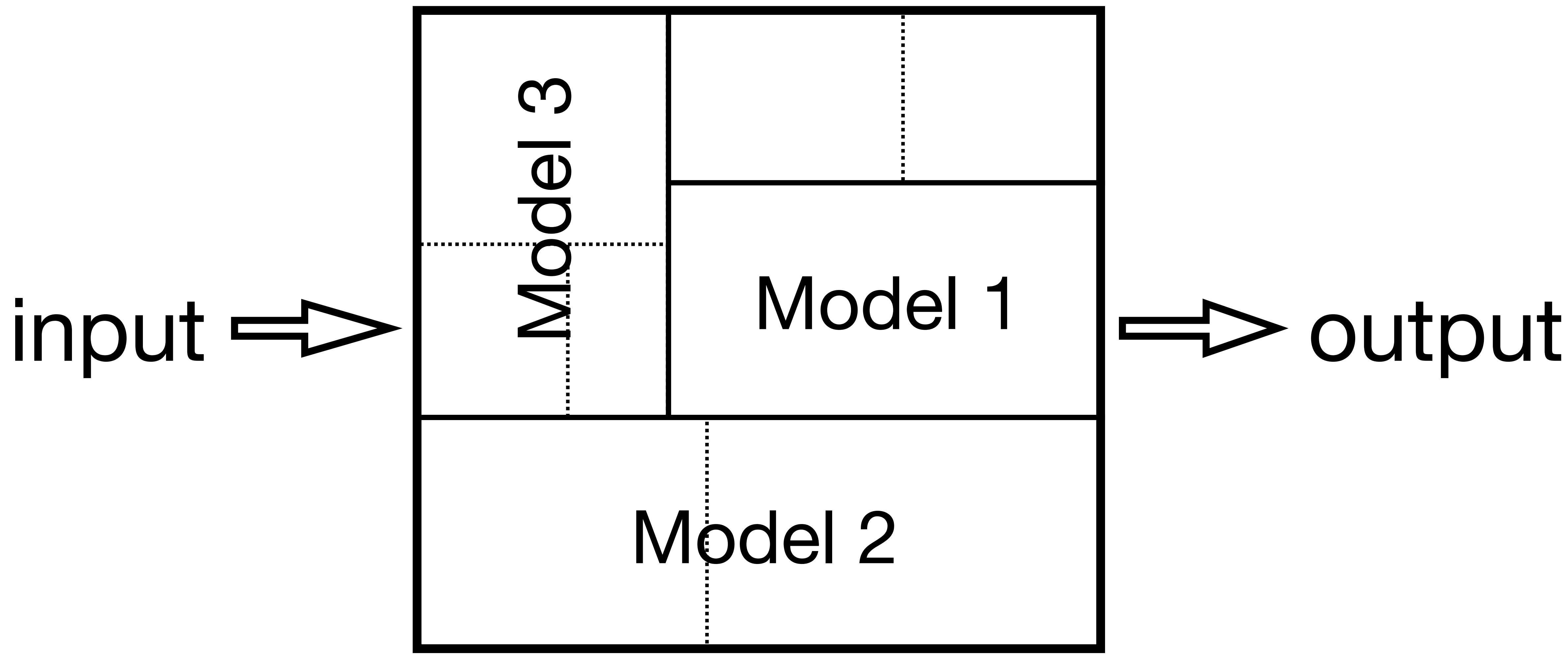
- How to link models across scales is not well understood physically or computationally
- We need to explore the “interfaces” between the scales, i.e., the numerical and physical interconnections
- Current codes are static and cumbersome
- We need new computational approaches to link models, which starts with a more flexible way to interconnect them

- Current codes based on monolithic designs
- Time consuming to introduce new models into codes, requiring them to have the same data structures, …
- Using the new model in another code would require the same type of modifications
- *Inefficient and a barrier to adopting new models*



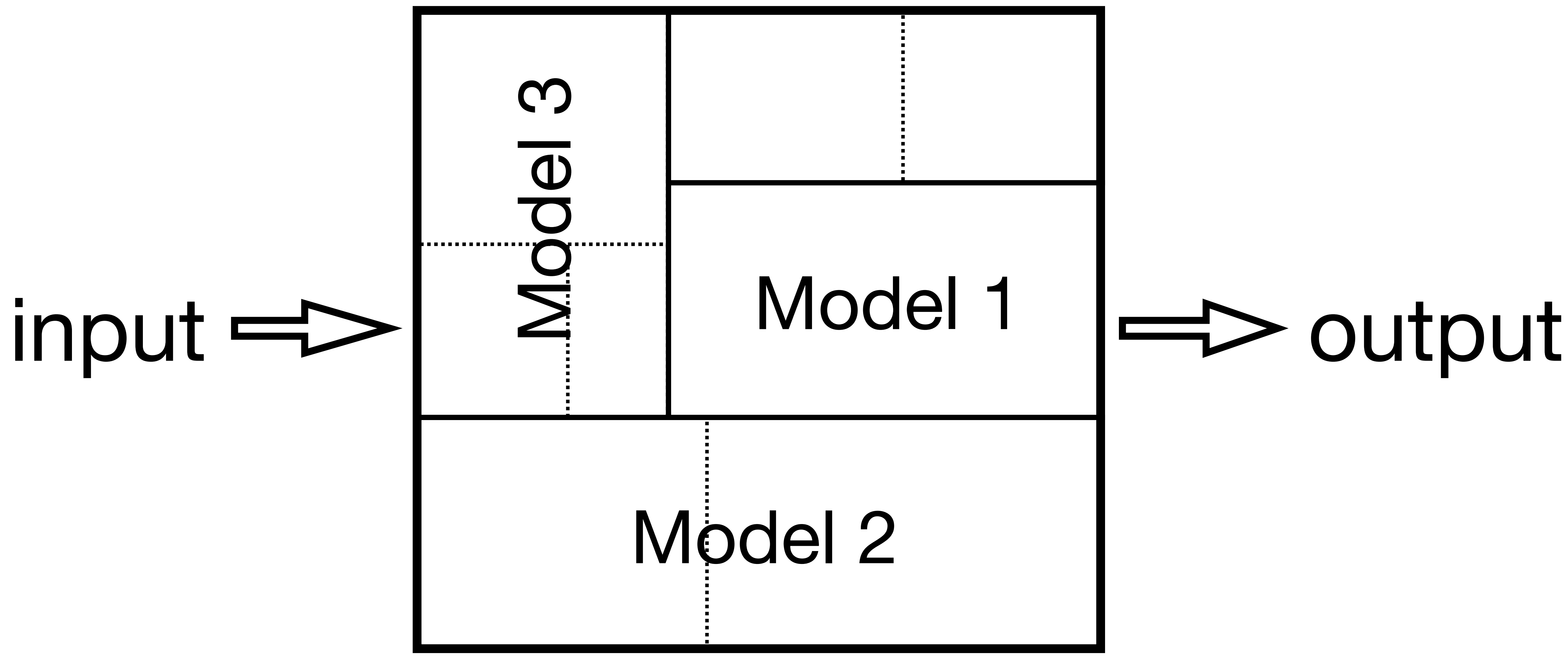
Challenge: current coding practice

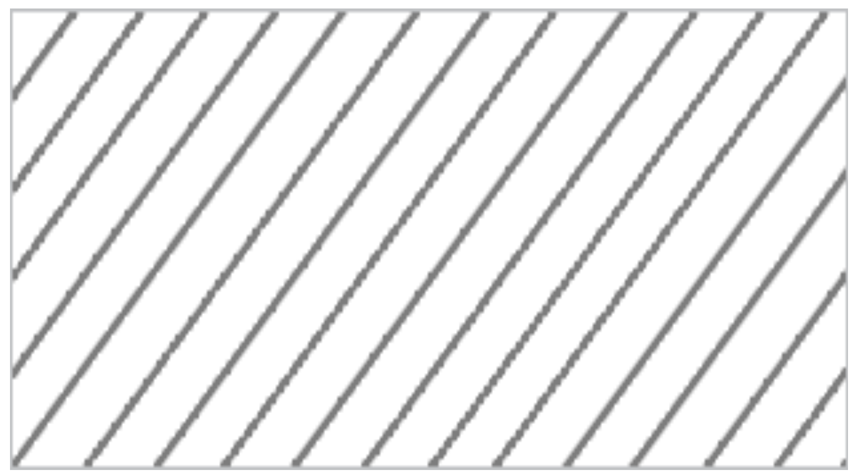
- Current codes based on monolithic designs
- Time consuming to introduce new models into codes, requiring them to have the same data structures, …
- Using the new model in another code would require the same type of modifications
- *Inefficient and a barrier to adopting new models*



Challenge: current coding practice

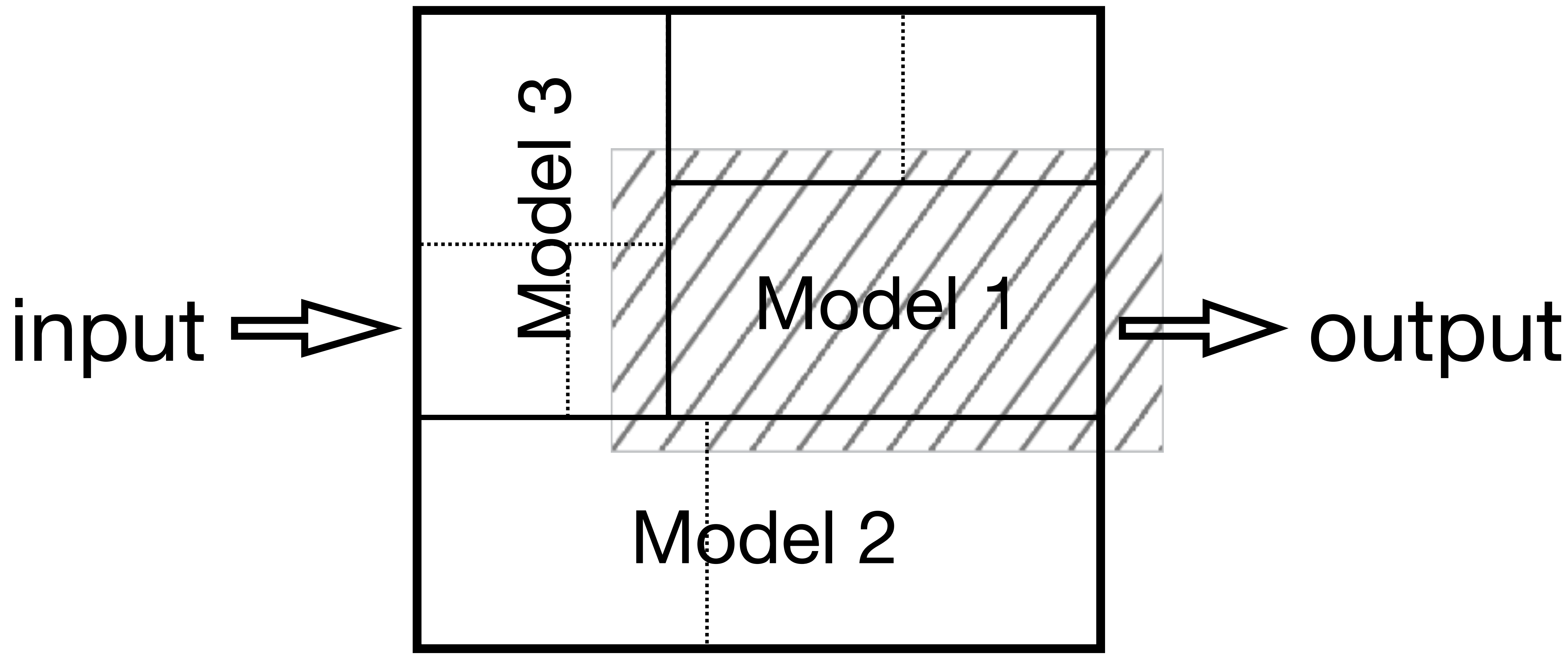
- Current codes based on monolithic designs
- Time consuming to introduce new models into codes, requiring them to have the same data structures, …
- Using the new model in another code would require the same type of modifications
- *Inefficient and a barrier to adopting new models*





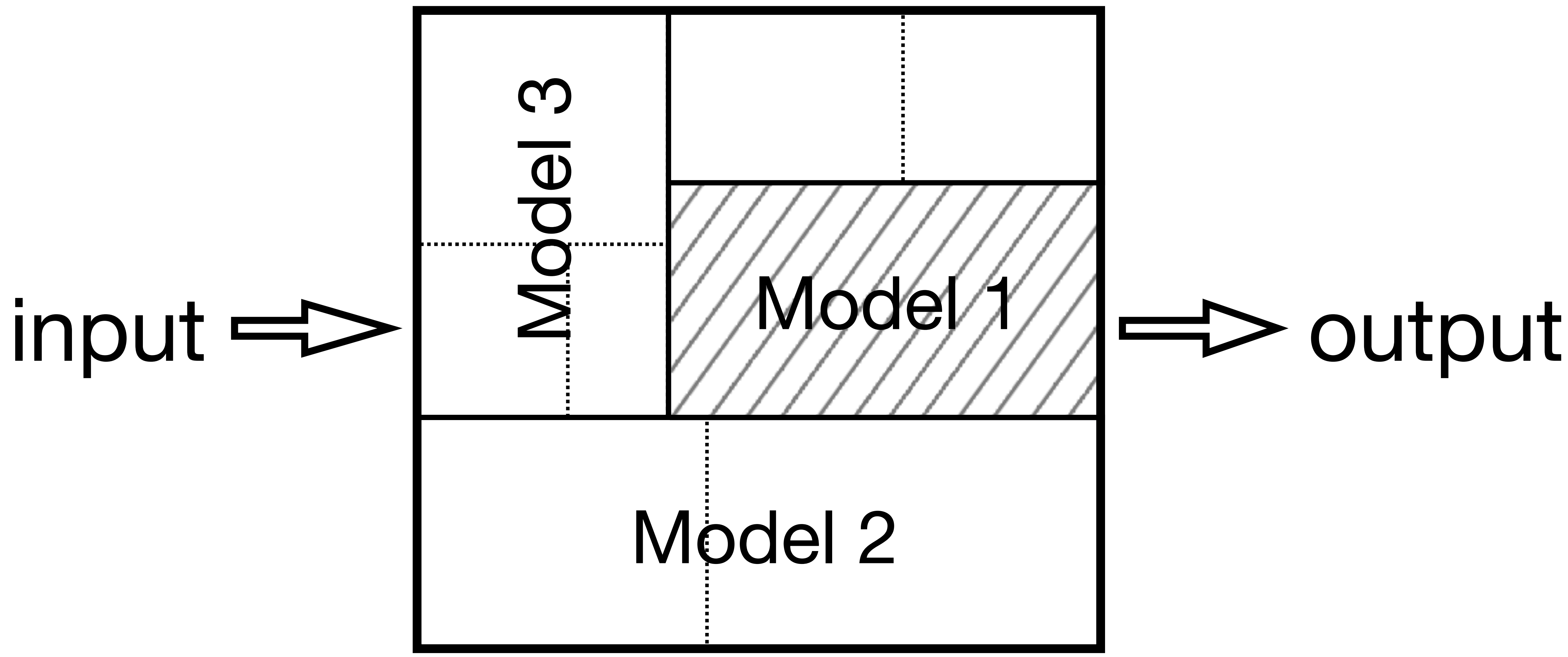
Challenge: current coding practice

- Current codes based on monolithic designs
- Time consuming to introduce new models into codes, requiring them to have the same data structures, ...
- Using the new model in another code would require the same type of modifications
- *Inefficient and a barrier to adopting new models*

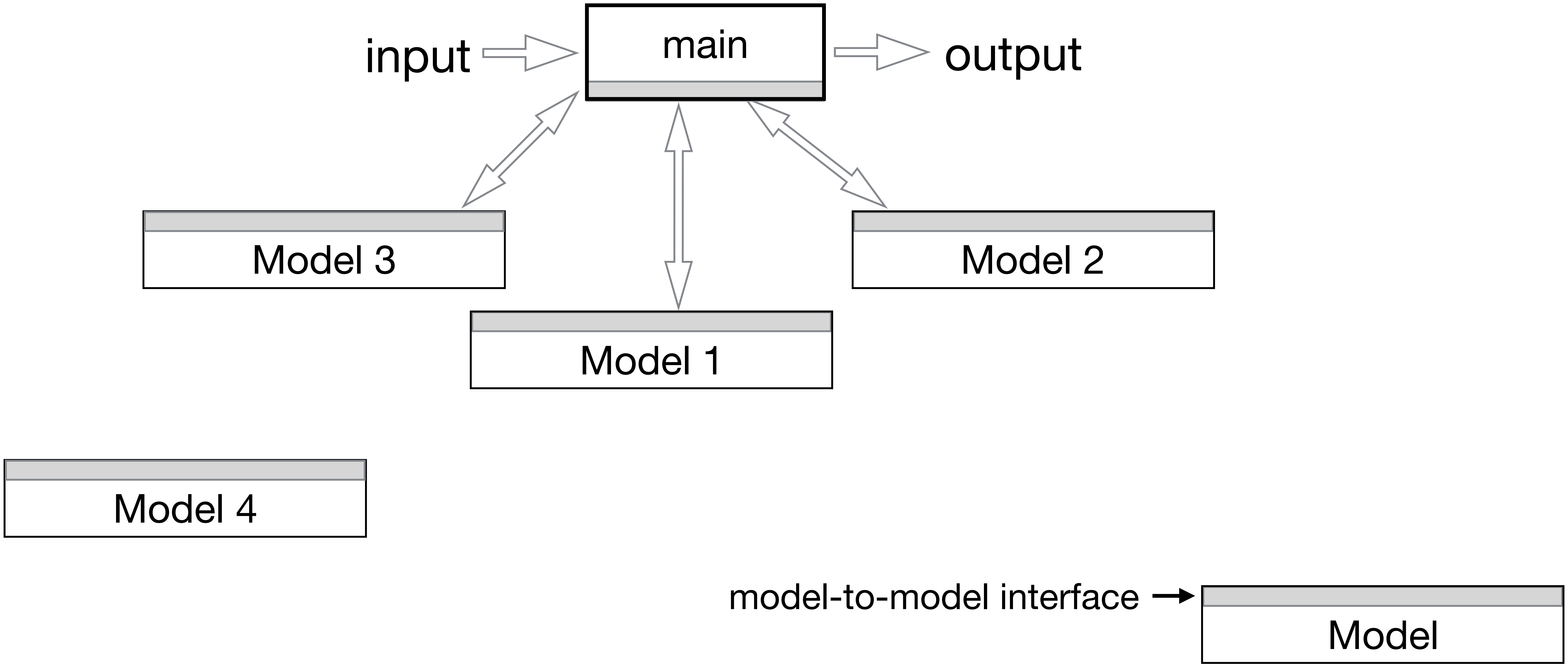


Challenge: current coding practice

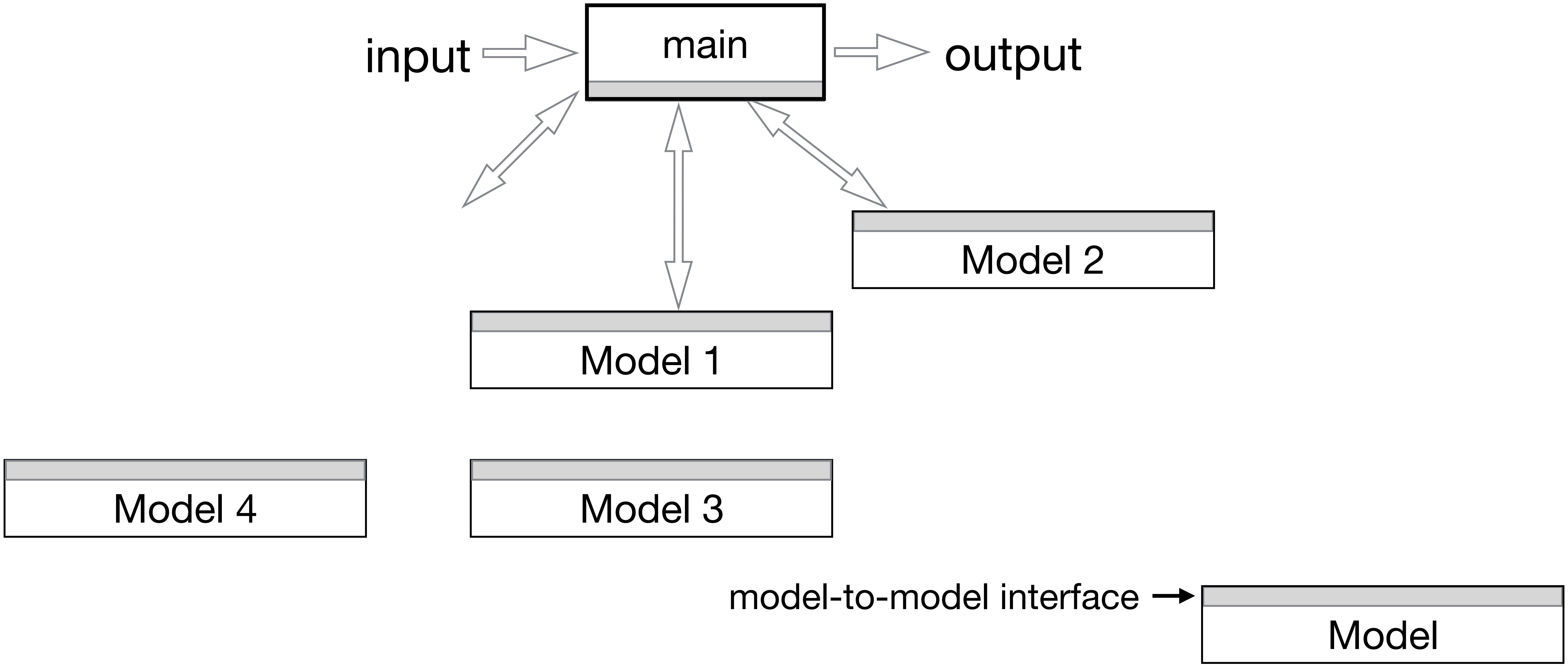
- Current codes based on monolithic designs
- Time consuming to introduce new models into codes, requiring them to have the same data structures, ...
- Using the new model in another code would require the same type of modifications
- *Inefficient and a barrier to adopting new models*



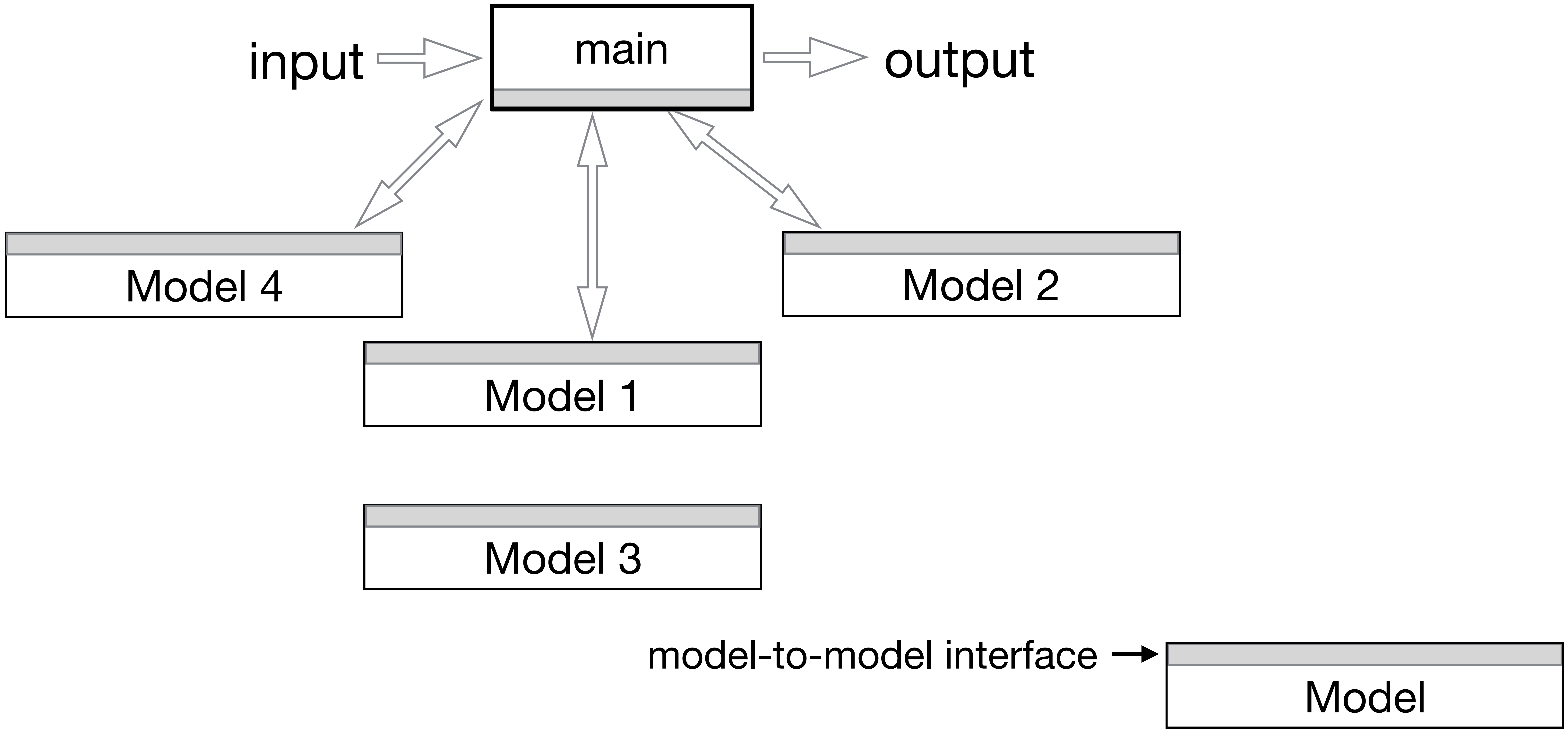
Challenge: current coding practice



Our goal: reusable and interchangeable models



Our goal: reusable and interchangeable models



Our goal: reusable and interchangeable models

We chose an approach taken by the Community Surface Dynamics Modeling System (CSDMS) centered at the University of Colorado

- hydrology across scales

CSDMS faces many of the same issues that we do in materials:

- using disparate models with disparate styles and computer languages developed by disparate groups located across the world
- need to link those models in an efficient way

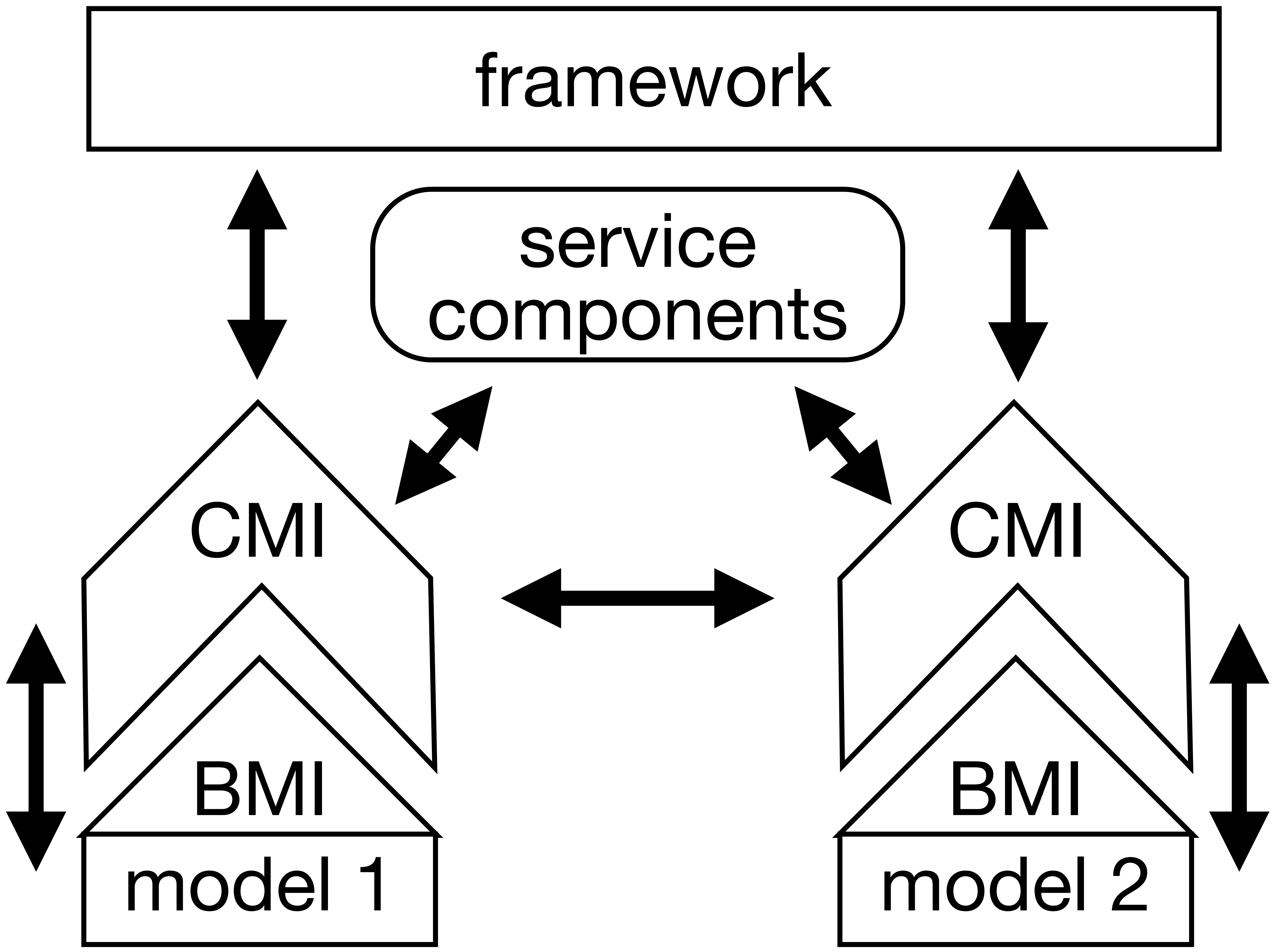
http://csdms.colorado.edu/wiki/Main_Page

Each model has added to it the Basic Model Interface (BMI)

The Common Model Interface (CMI) automatically handles conversions between languages (with Babel)

Their framework controls the calculation and the communication to/from models.

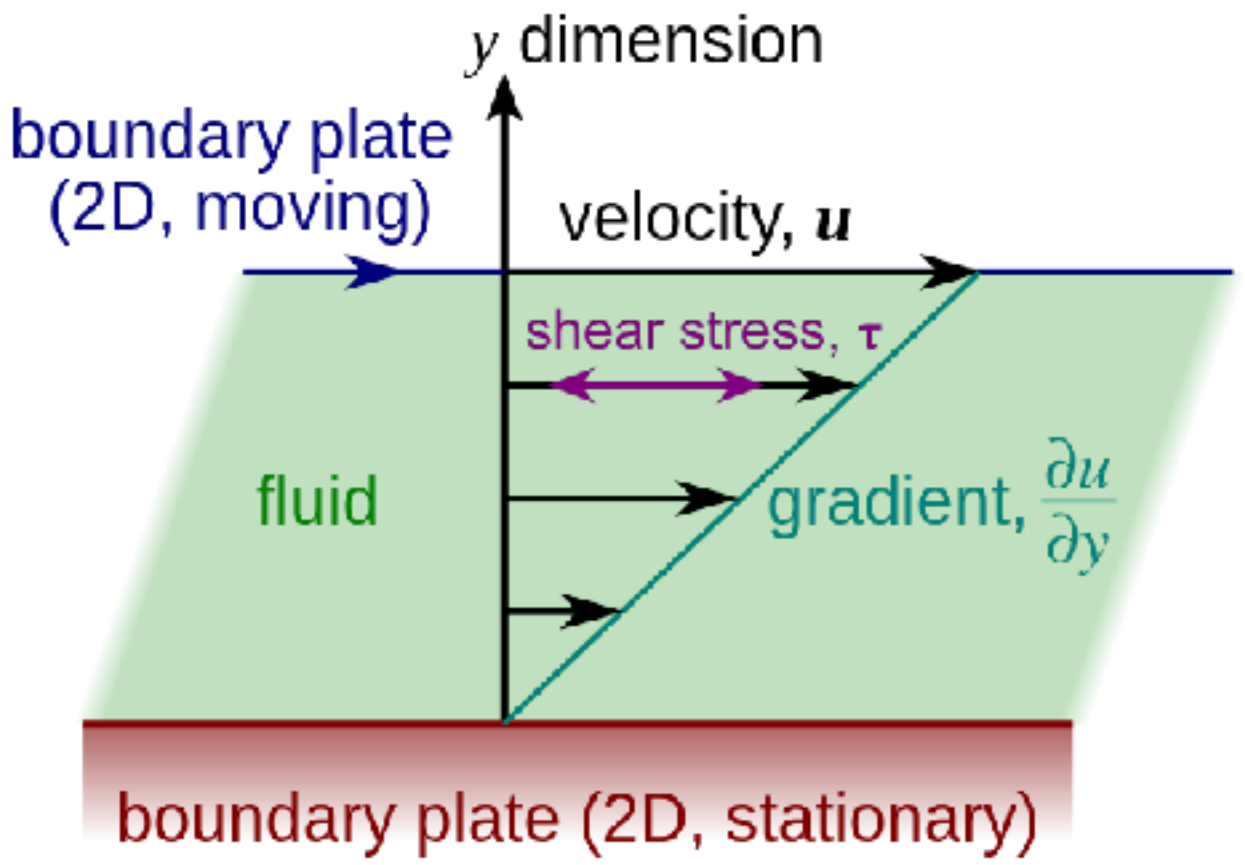
Designed to work on centralized, large-scale computers.

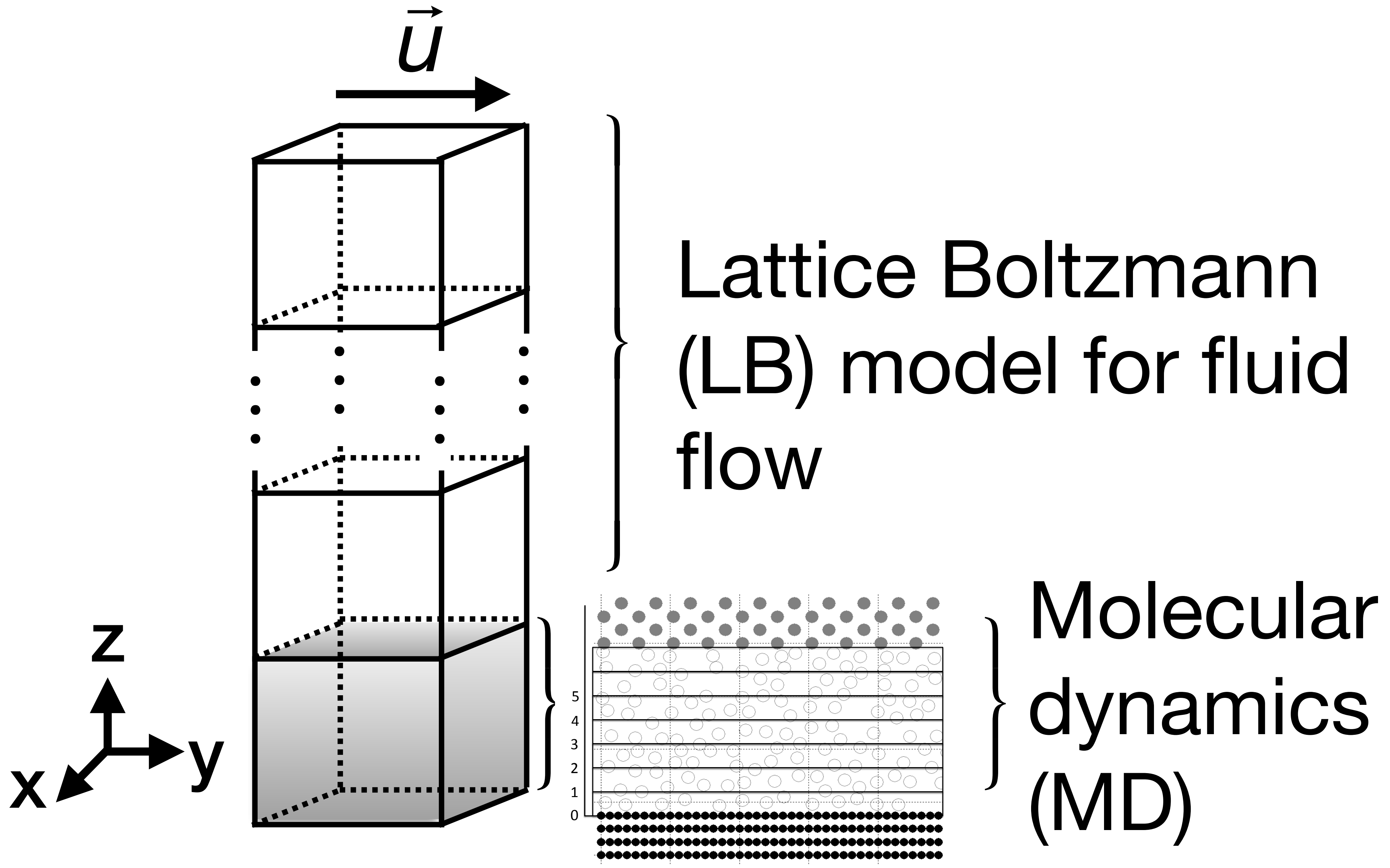


BMI/CMI model-to-model interface

We linked fluid flow calculations (Lattice-Boltzmann) to atomistics (molecular dynamics) to calculate effects of atomic interactions with the surface on the slip velocity at boundary plate in Couette flow

- Added a BMI to each model
- Linked models through the BMI
- Used a simple iterative solution to find the slip-velocity as a function of atomic interactions, including surface structure

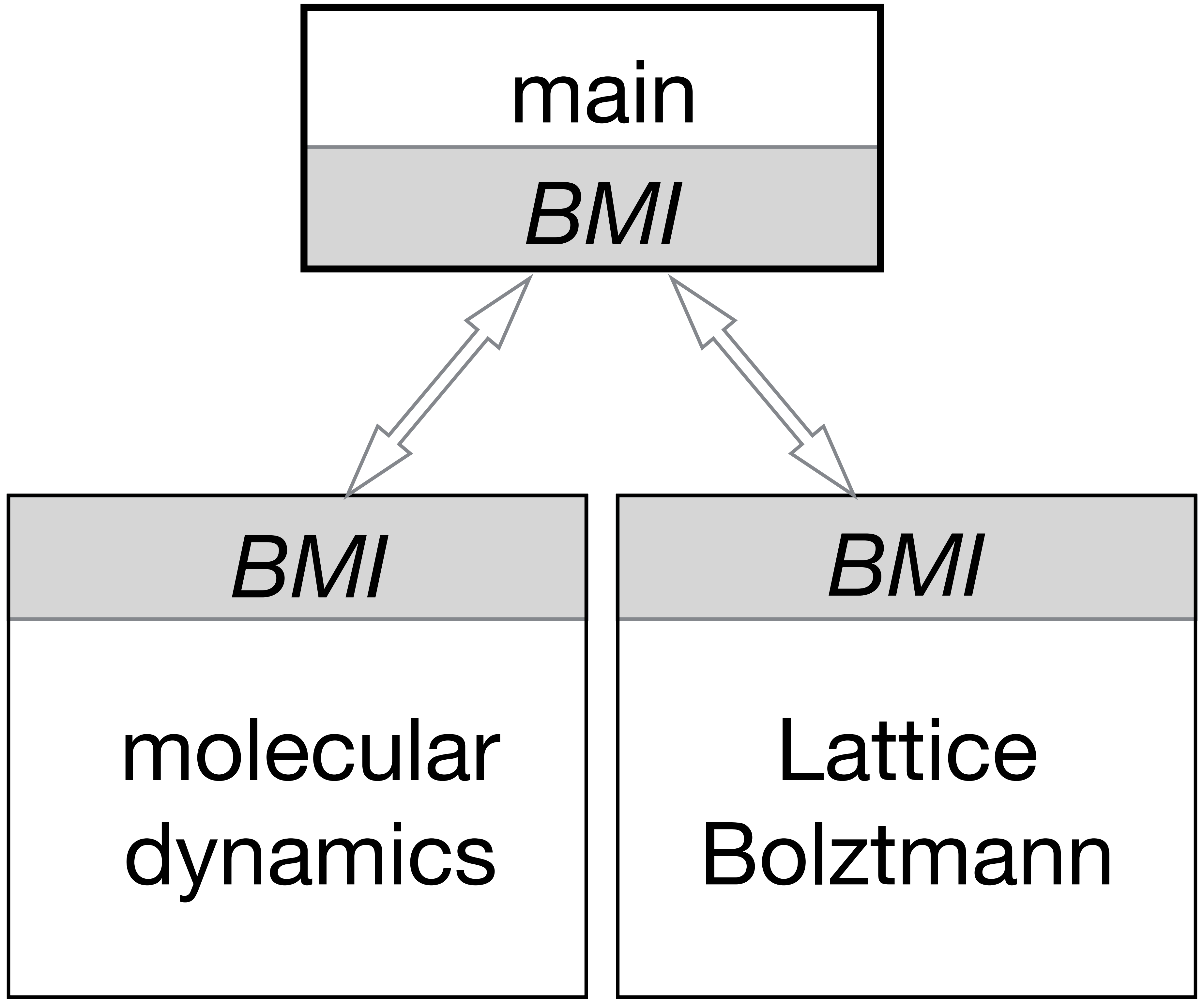




Example materials problem

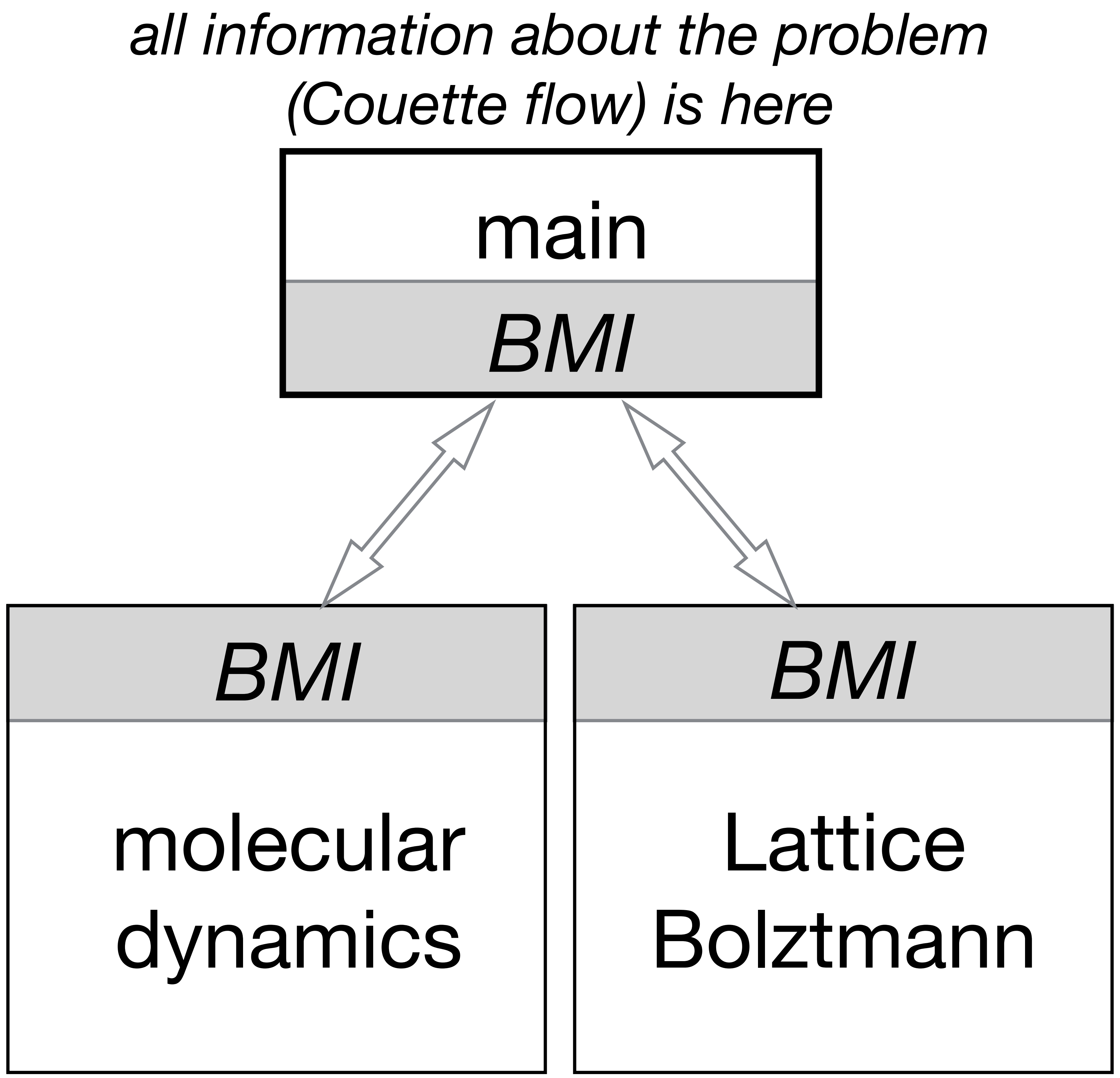
“A model-to-model interface for multiscale materials simulations,” P. E. Antonelli, K. M. Bryden, and R. LeSar, Computational Materials Science 123, 244-251 (2016)

- The Lattice Boltzmann (LB) and molecular dynamics (MD) models were autonomous
- Each model had its own internal units and data structure
- Each model was solved with its own time step (very different in size)
- Each model had its own implementation of boundary conditions
- Each model had its own requirements for convergence



Information mediation between LB and MD

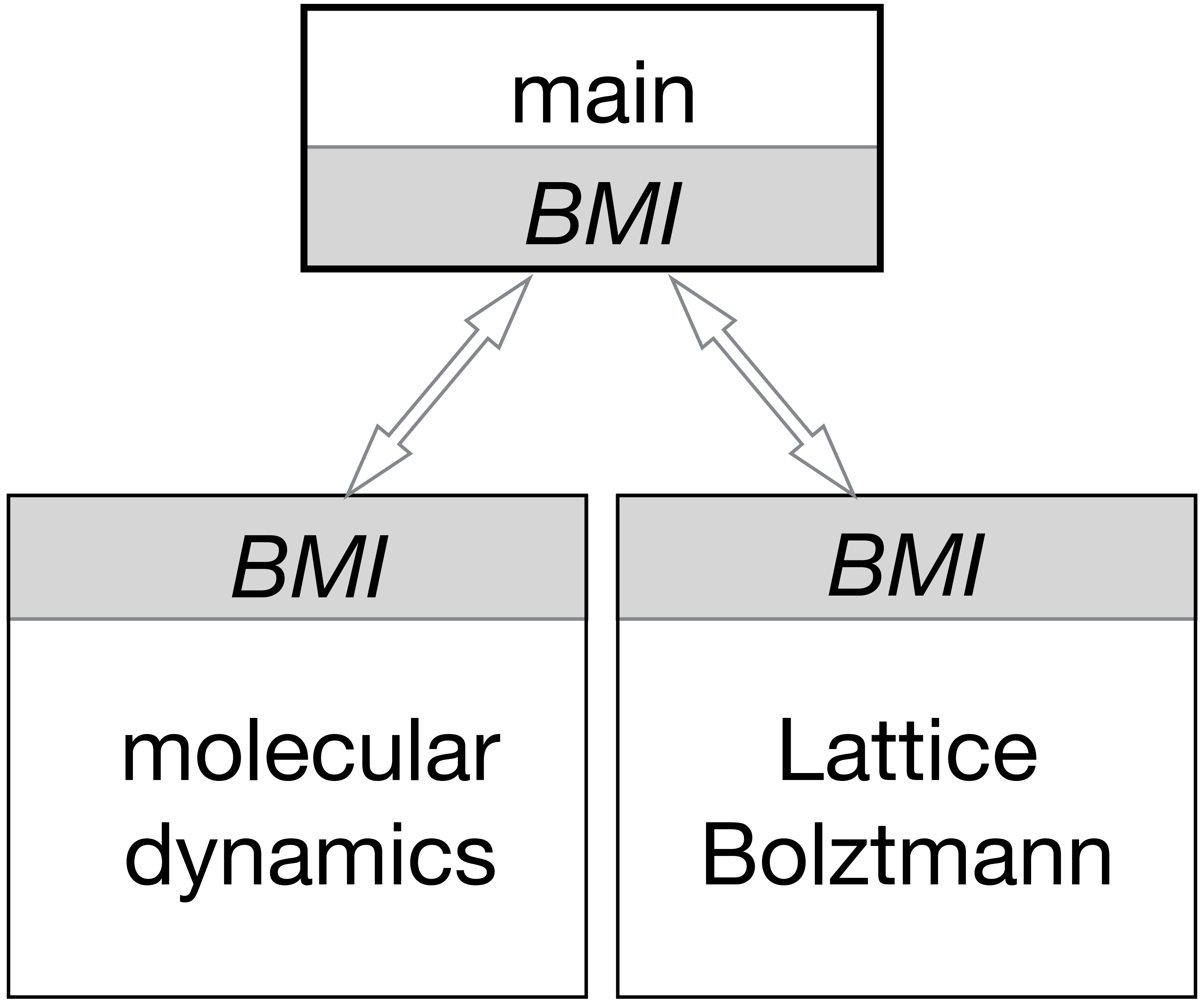
- The Lattice Boltzmann (LB) and molecular dynamics (MD) models were autonomous
- Each model had its own internal units and data structure
- Each model was solved with its own time step (very different in size)
- Each model had its own implementation of boundary conditions
- Each model had its own requirements for convergence



Information mediation between LB and MD

- The Lattice Boltzmann (LB) and molecular dynamics (MD) models were autonomous
- Each model had its own internal units and data structure
- Each model was solved with its own time step (very different in size)
- Each model had its own implementation of boundary conditions
- Each model had its own requirements for convergence

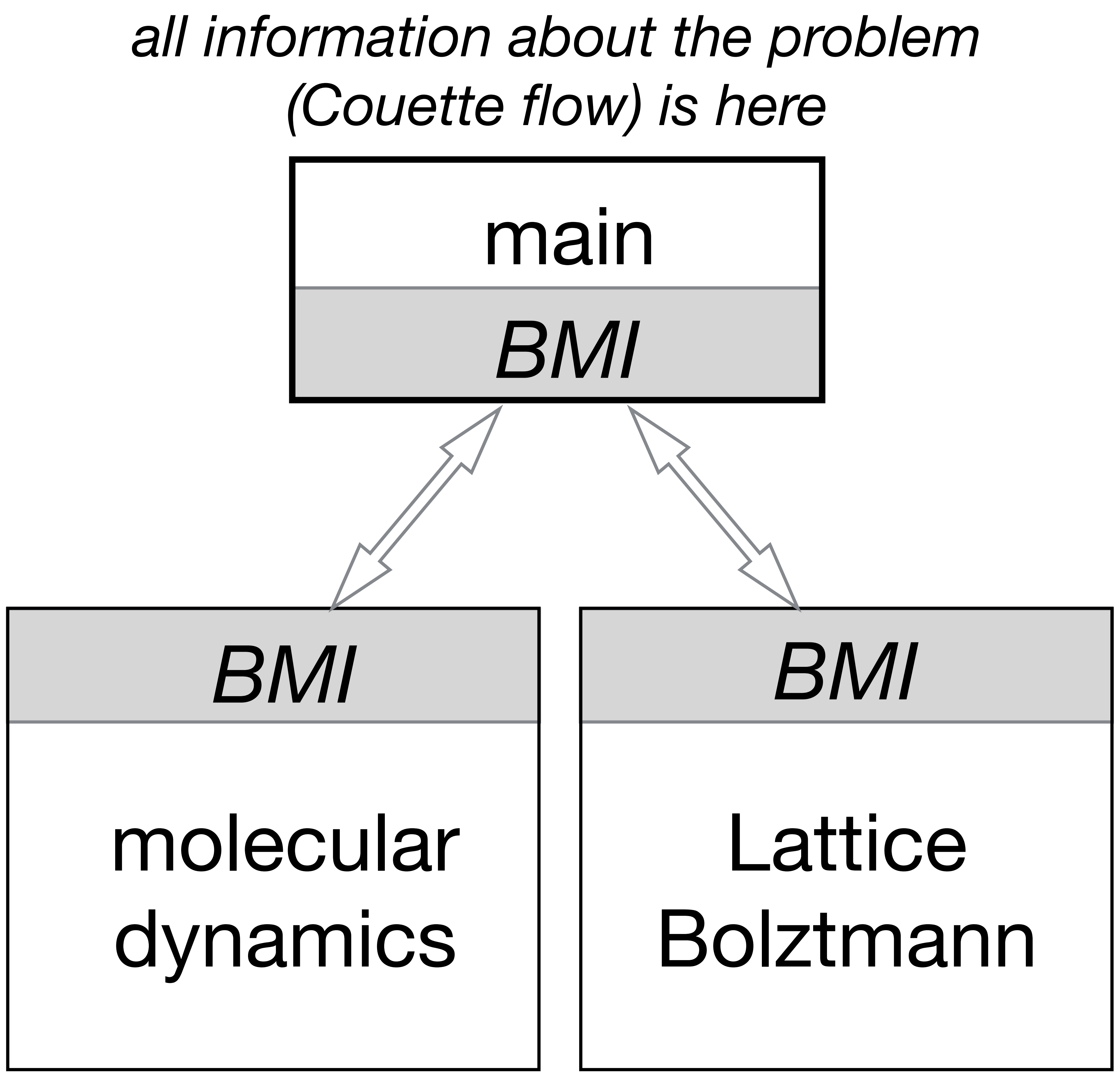
all information about the problem (Couette flow) is here



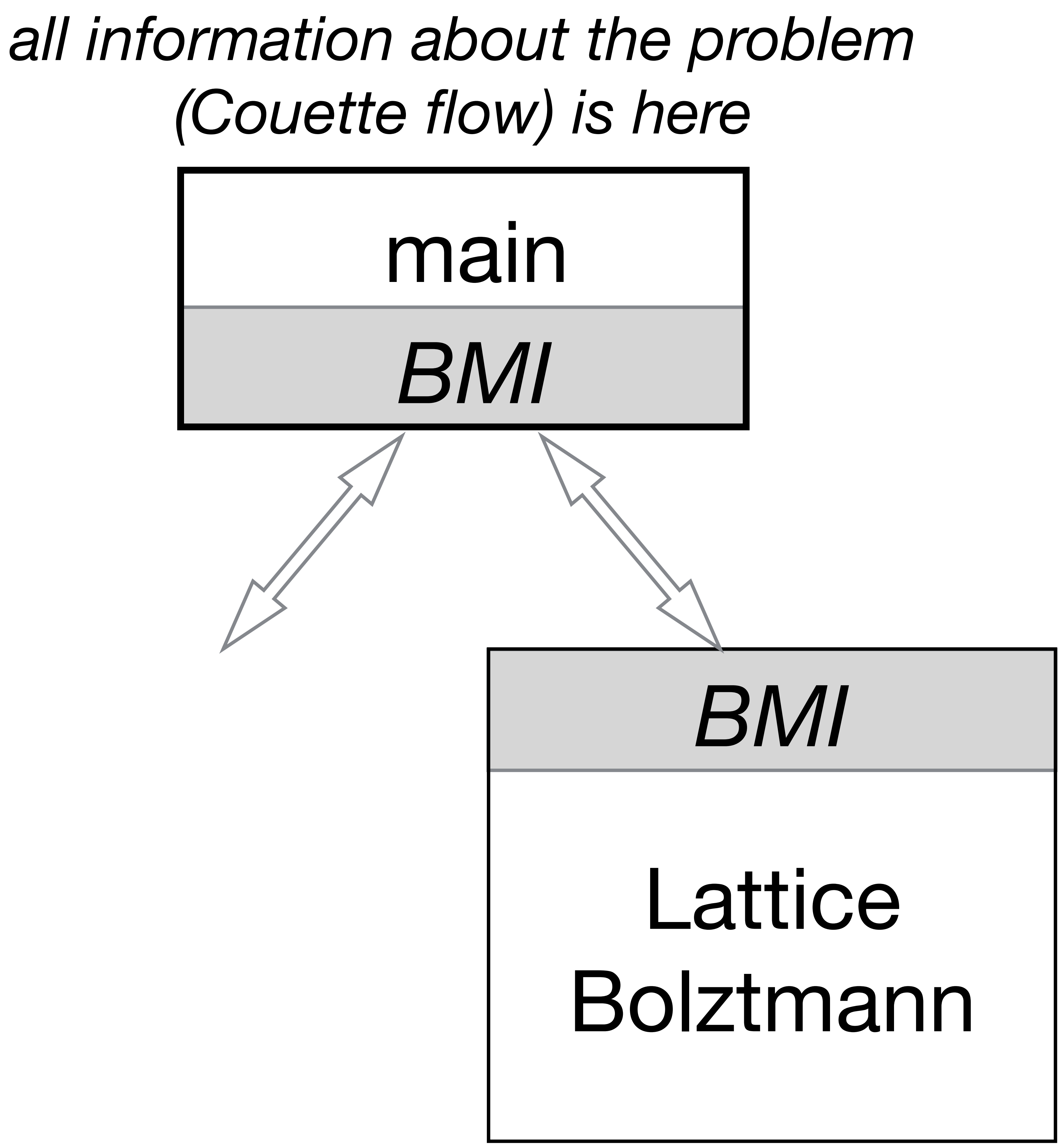
The BMI handles all communication with the models: input, instructions, output, … through a standard set of functions.

Information mediation between LB and MD

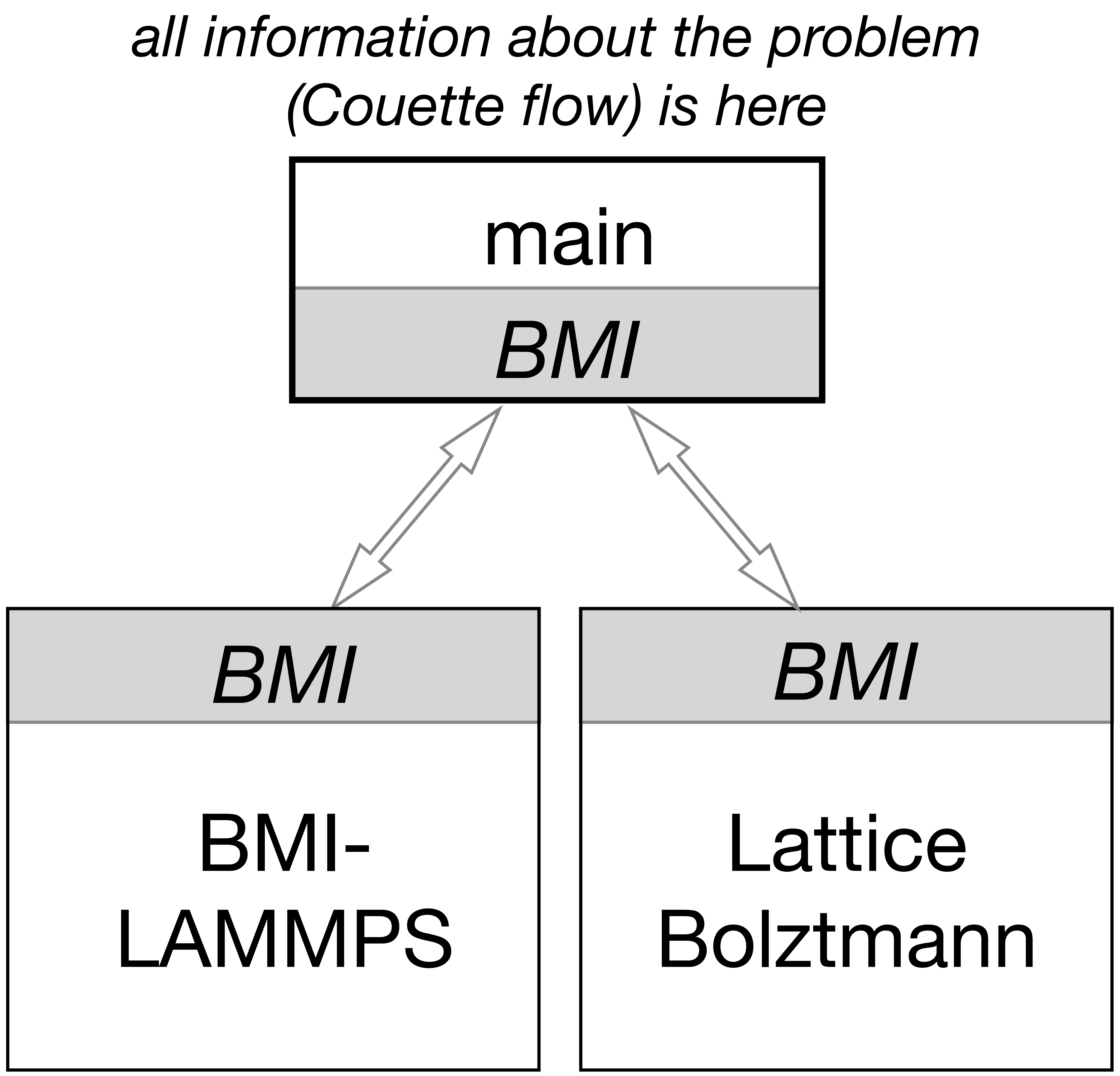
- Goal is to add the BMI to codes widely used for materials modeling but not developed by us. First step:
    - LAMMPS: an open-source molecular dynamics program created at Sandia
- Added the BMI to LAMMPS: BMI-LAMMPS
- Linked BMI-LAMMPS to our Lattice-Boltzmann code with no additional changes to either code
- Establishes the utility of this approach in computational materials modeling



- Goal is to add the BMI to codes widely used for materials modeling but not developed by us. First step:
  - LAMMPS: an open-source molecular dynamics program created at Sandia
- Added the BMI to LAMMPS: BMI-LAMMPS
- Linked BMI-LAMMPS to our Lattice-Boltzmann code with no additional changes to either code
- Establishes the utility of this approach in computational materials modeling



- Goal is to add the BMI to codes widely used for materials modeling but not developed by us. First step:
    - LAMMPS: an open-source molecular dynamics program created at Sandia
- Added the BMI to LAMMPS: BMI-LAMMPS
- Linked BMI-LAMMPS to our Lattice-Boltzmann code with no additional changes to either code
- Establishes the utility of this approach in computational materials modeling



CSDMS is based on the use of a large-scale centralized computing facility

- the distributed set of researchers are part of one large project and the linked models are run at the CSDMS facility

The materials modeling community is not a single project and has no single shared computer facility

- researchers run on a distributed set of computers
- need a different approach

Cloud-based computing will enable collaborative projects between disparate model developers

Limitation of CSDMS approach

- Does not require centralized computational facilities
- Uses models as web services (a service that is offered by an electronic device to another electronic device, communicating with each other via the web)
- Developed applications based on a collection of independent microservices
  - microservices are stateless, fine-grained, easily replaced, independently deployable, …
  - we are creating microservices of models with the BMI/CMI interface

- Decomposing an application into different smaller microservices improves modularity and makes the application easier to understand, develop and test

- Parallelizes code development by enabling small autonomous teams to develop, deploy and scale their respective services independently.

- Allows the architecture of an individual service (e.g., application) to emerge through continuous restructuring of code

- Enables multiple instantiations of models that run simultaneously and continuously

- Enables identification (with metadata) of each model, attribution to author, version control, …

Benefits of microservice-based architecture

http://ec2-34-216-89-124.us-west-2.compute.amazonaws.com:5000/

- Developing ways to streamline creation of BMI-based microservices
- Adding the BMI/CMI to models that will be linked for specific applications (will be developed as microservices)
- Developing strategy for applications based on linked microservices
- Optimizing the “interfaces” between the models, i.e., the numerical and physical interconnections
- Will compare application development time and computational speed of linked model applications with traditional application development and use

Current and future work

Vision

Develop a library of multiscale materials models, created by the FE modeling community and implemented as microservices, to enable FE to create dynamic simulation tools in support of affordable, low carbon, high efficiency, advanced power systems.

Will enable FE to establish leadership in an emerging area of computation.

LB fluid flow (FF)

MD 1

FD heat flow (HF)

LB FF/HF

BMI-LAMMPS

⋮

Entries in library could be computational models/simulations, databases, or …

Questions?