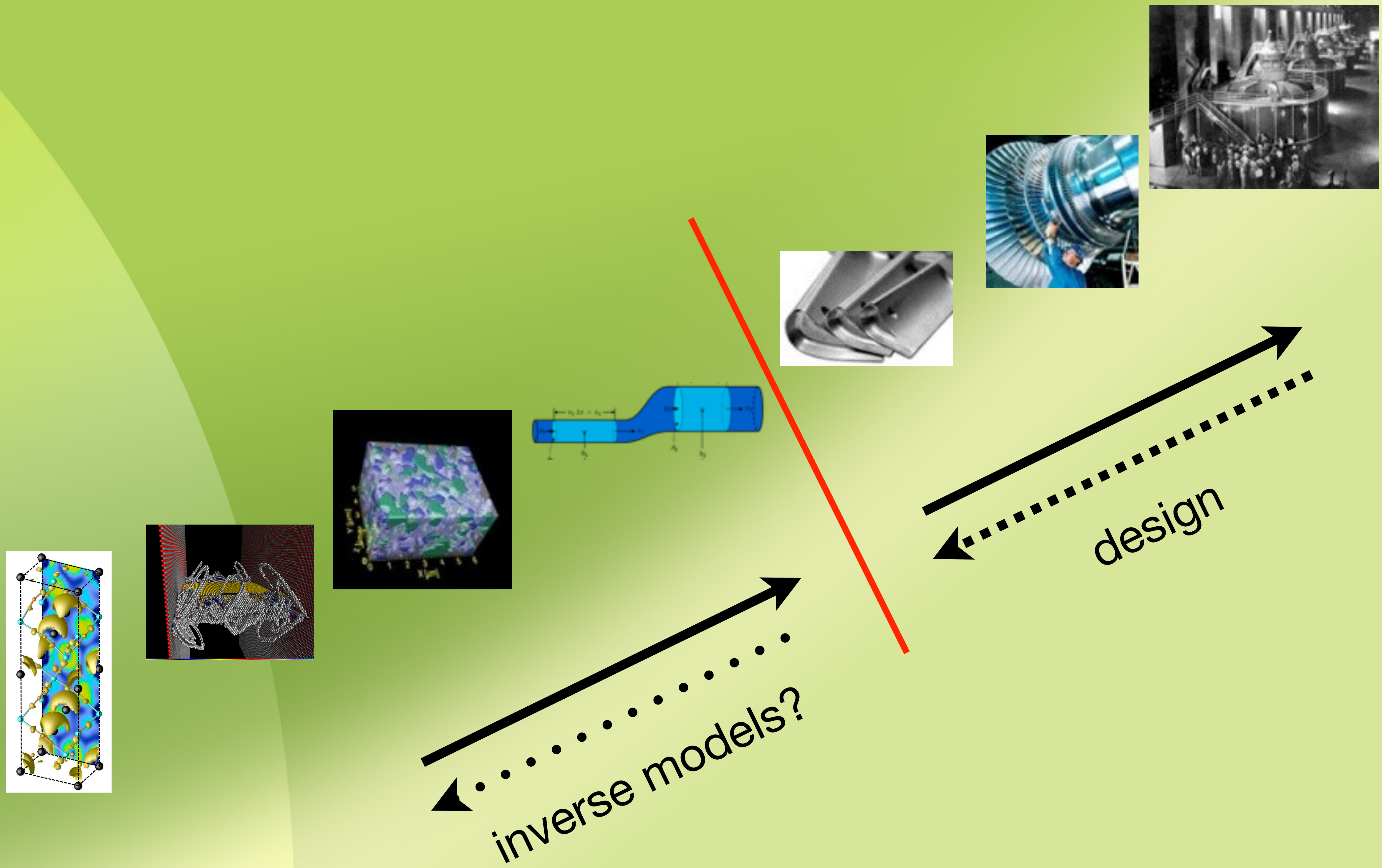
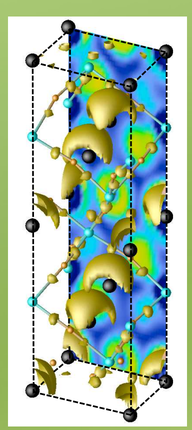
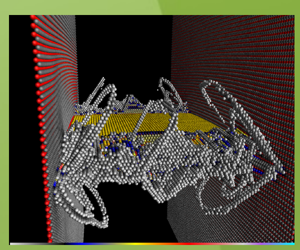
Implementing a basic model interface to support the rapid use of materials models in design

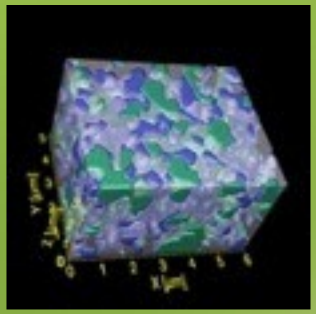


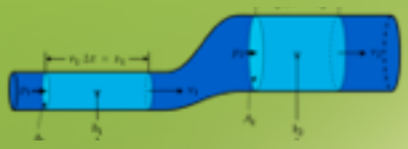
Design across scales

“information passing”, in which information at one scale is passed as input to a larger scale









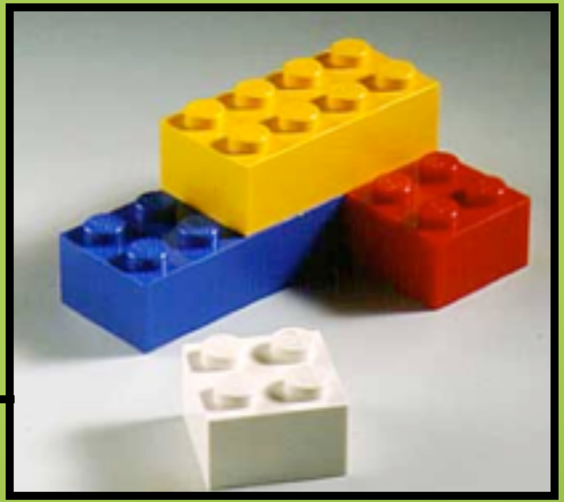
“concurrent multiscale”, which is used when separation of scales is not possible or desirable

- models
- simulations
- databases
- sensors
- ...

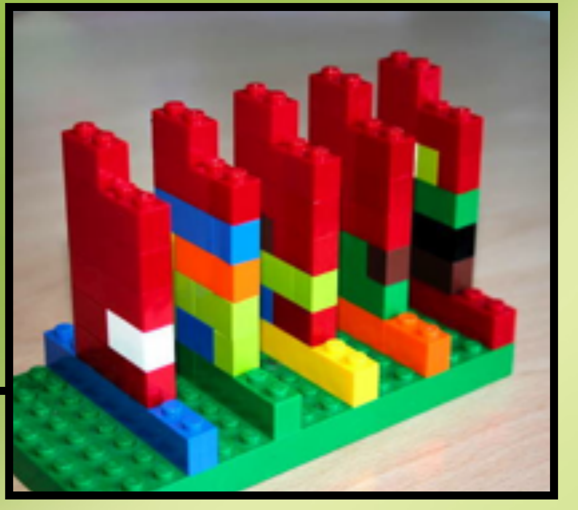
- models
- simulations
- databases
- sensors
- ...

“models”

Snap



Build



Do

New decision making paradigm

Decision making environments that integrate all the information, models, and other artifacts related to a product or process.

What's needed

1. Integration
2. Mediation
3. Interaction

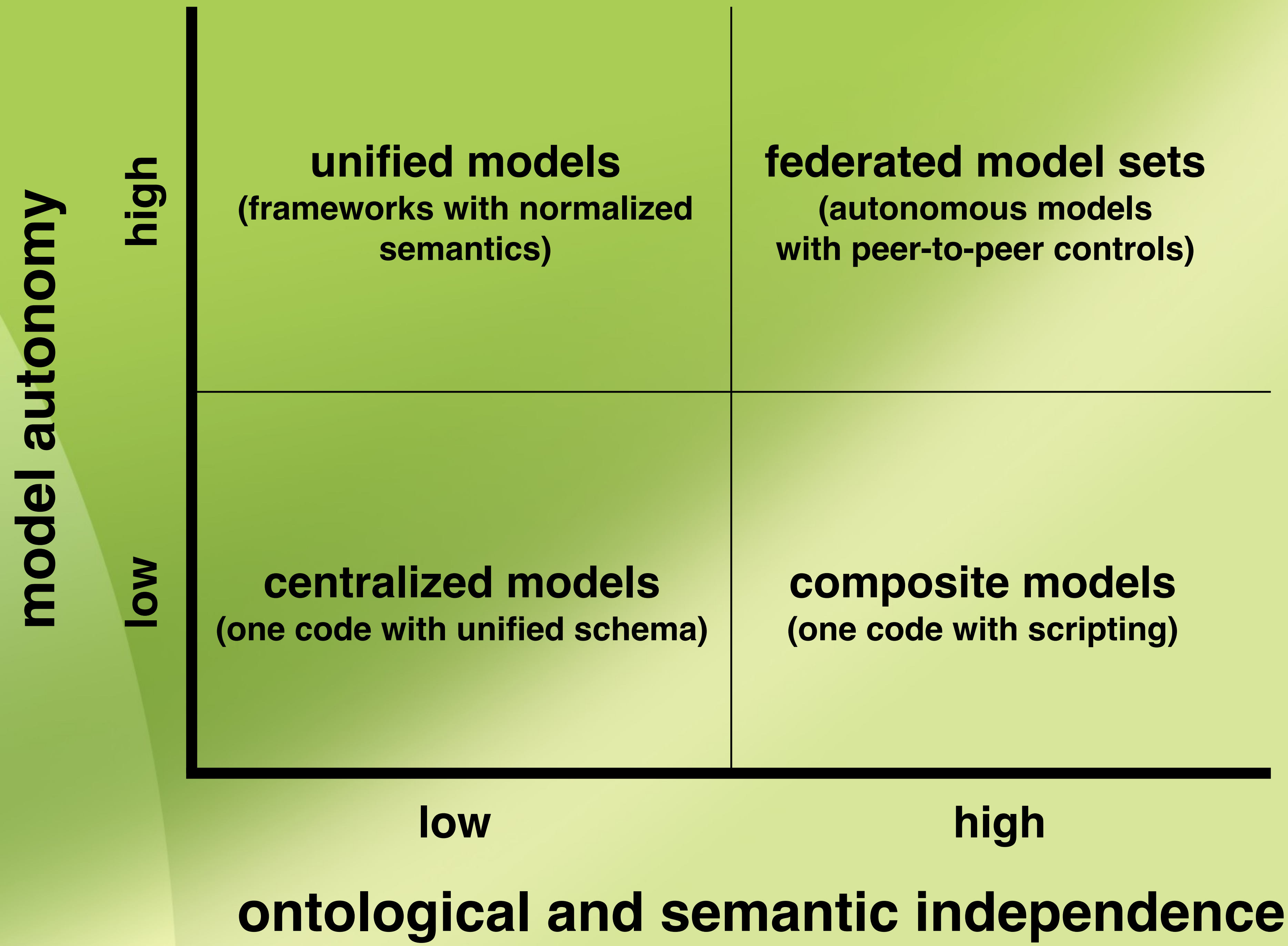
What's needed

1. Integration
2. **Mediation**
3. Interaction

Enables disparate models to effectively communicate and work together in support of engineering decisions

"… a centralized model encompassing a set of other models"

- integration framework
- global ontology and semantics



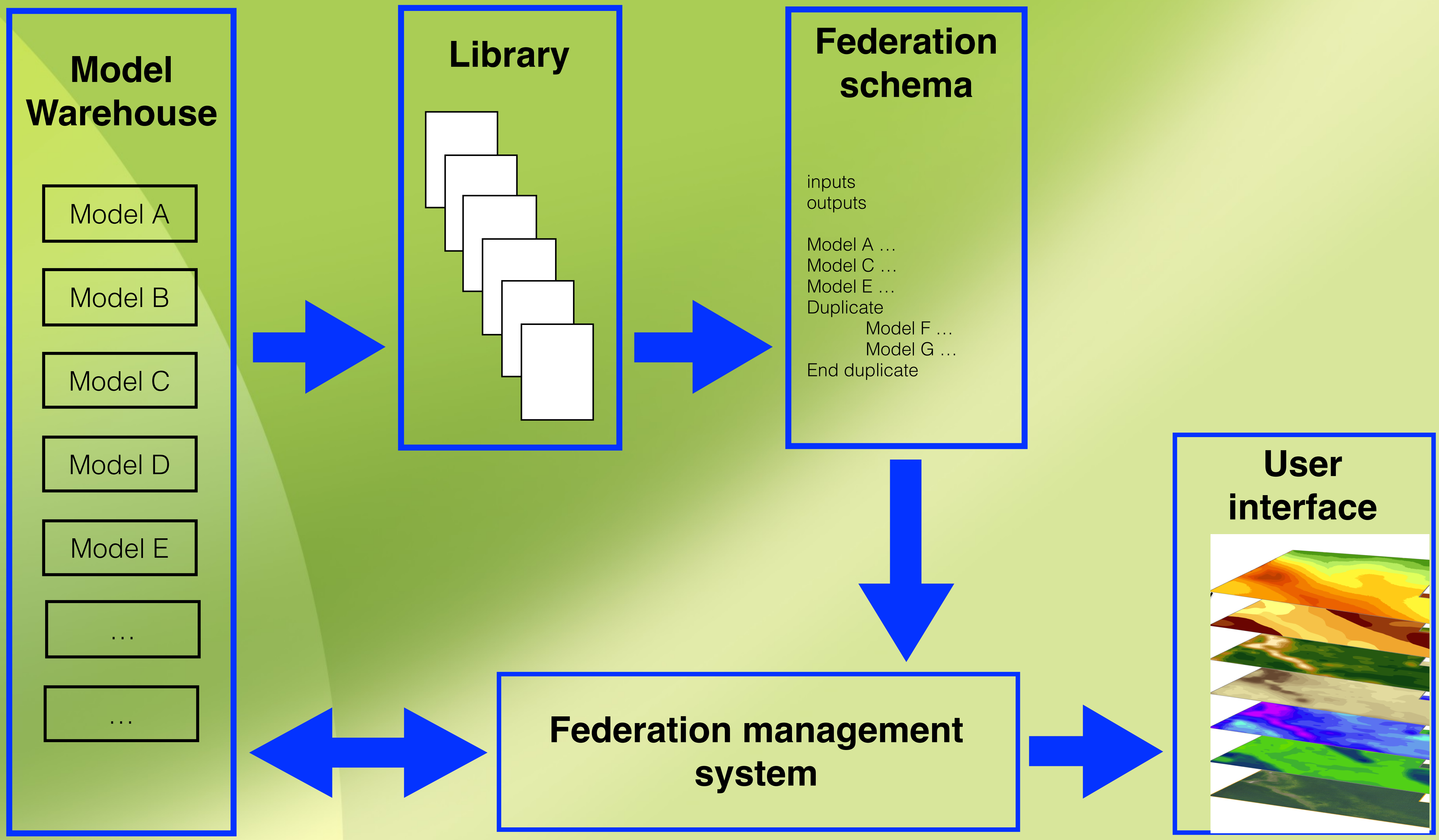
Model portability

To provide

- high degree of independence for component models;
- a common, light-weight mechanism for model linkage; and
- a basis for deploying the federated model set.

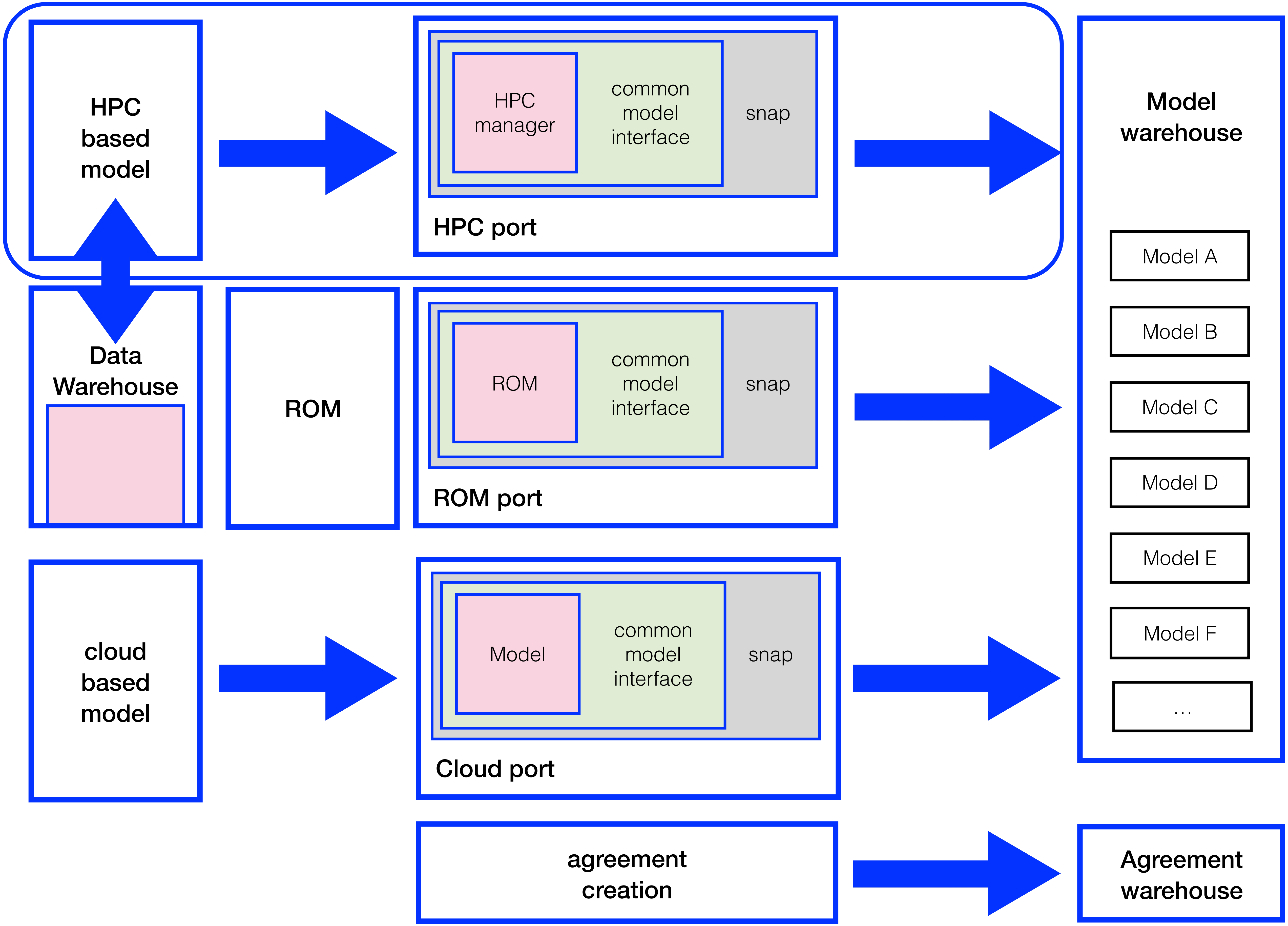
To provide

- high degree of independence for component models;
- a common, light-weight mechanism for model linkage; and
- a basis for deploying the federated model set.

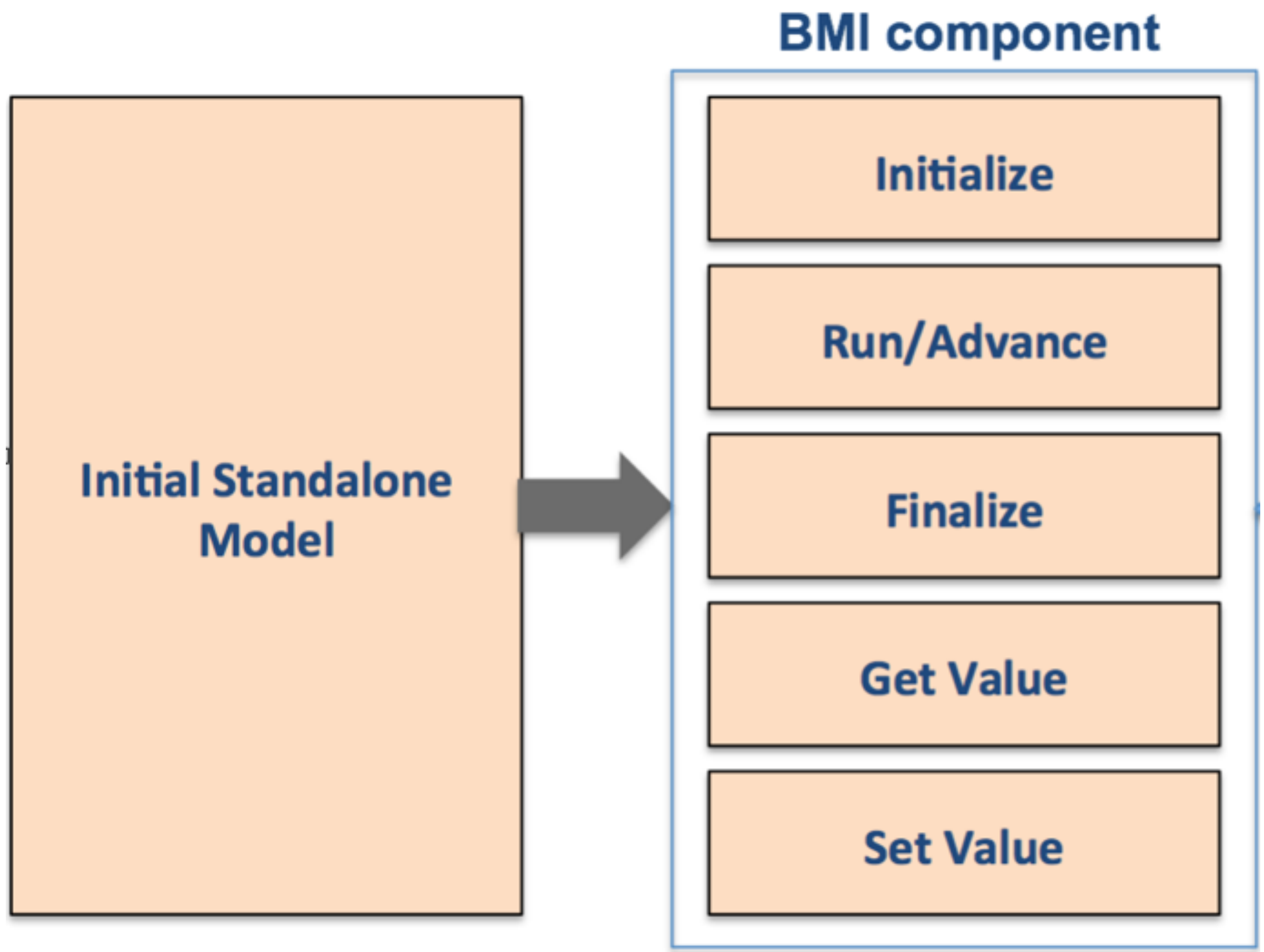


Components and information flow

- Content creation process
- Model integration schema
- ROM management schema
- Federation management system
- Development of a domain specific language

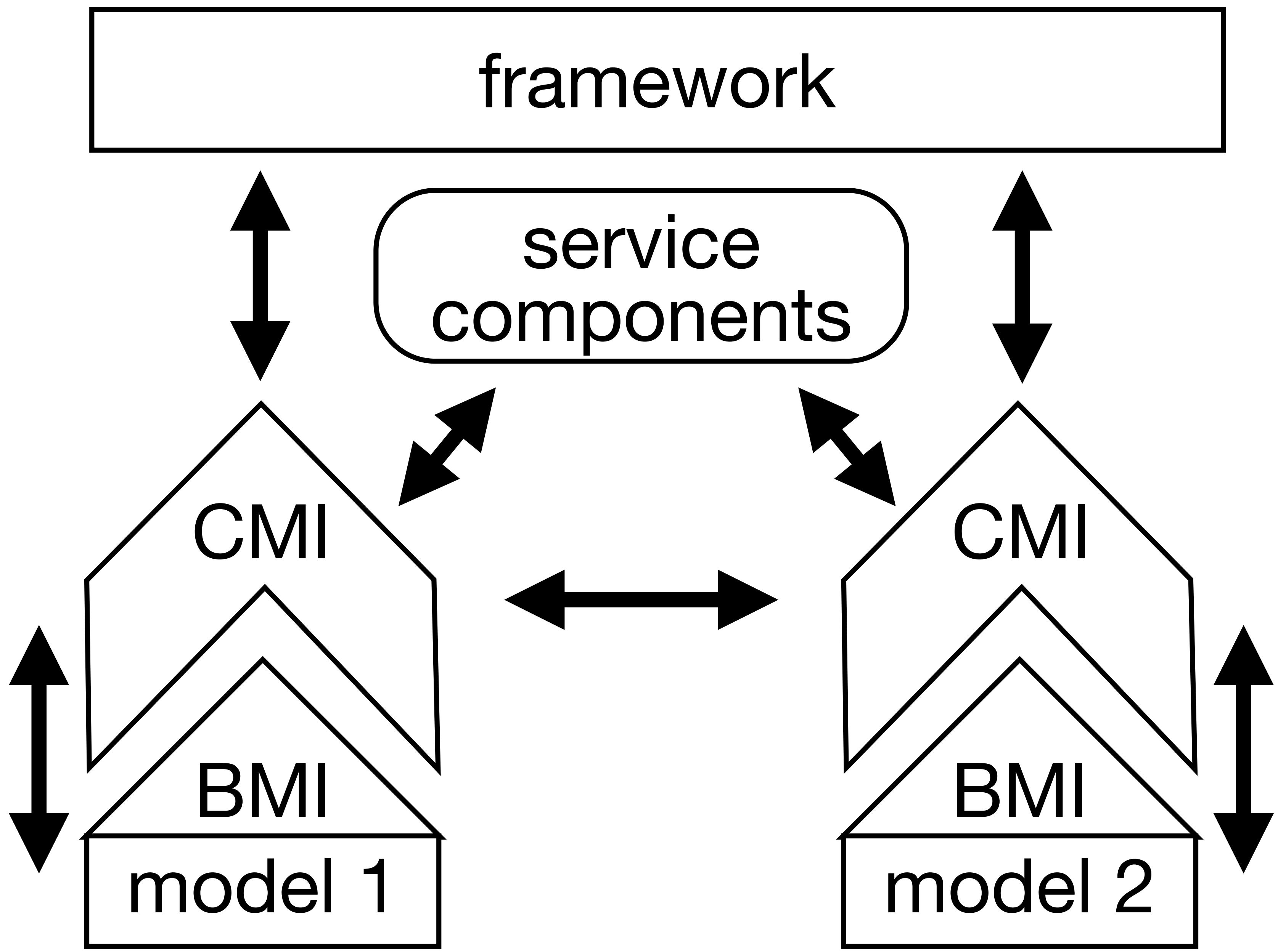


User defined content creation process

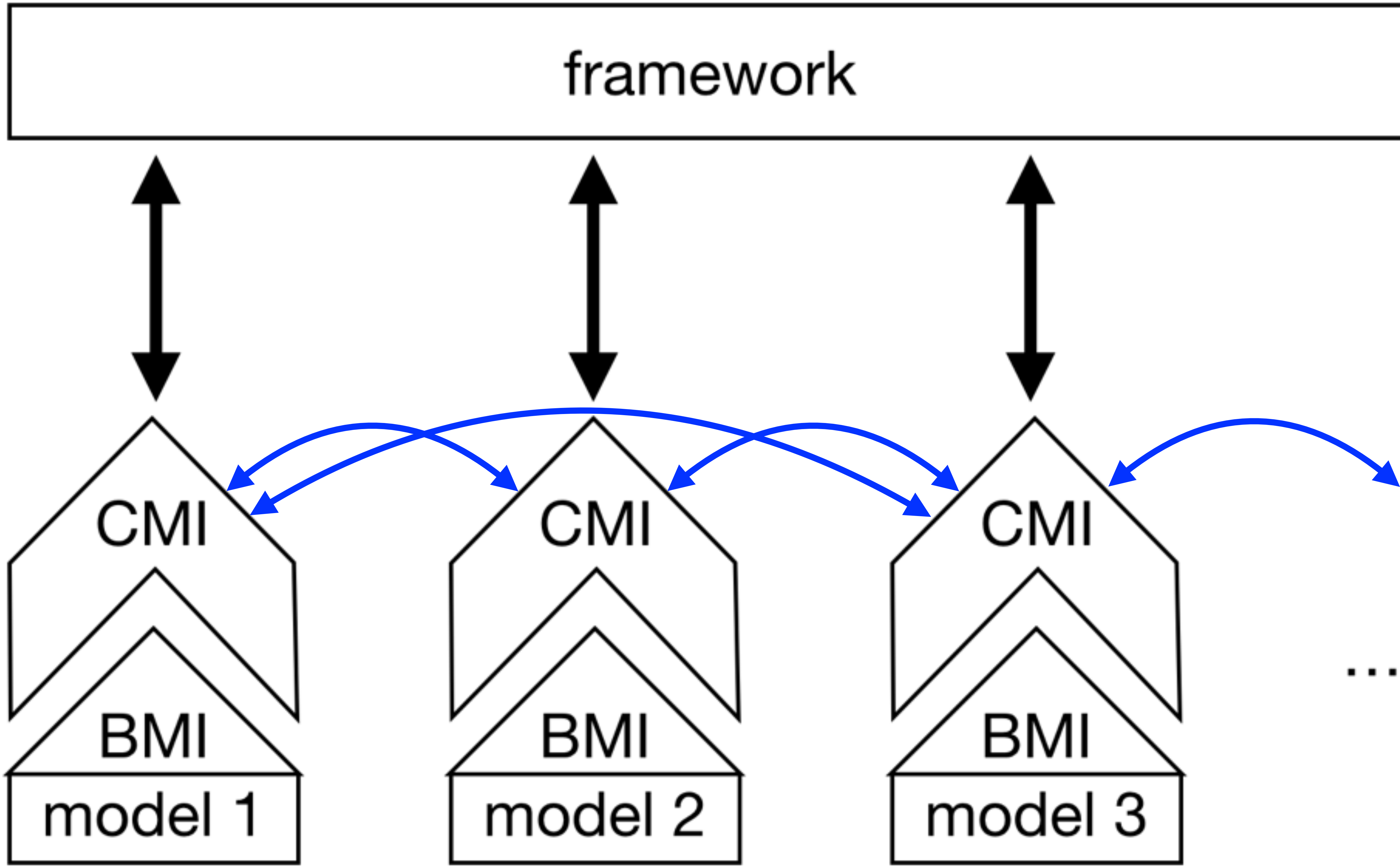


initial model integration schema

(University of Colorado: Community Surface Dynamics Modeling System)



BMI - CMI - Framework



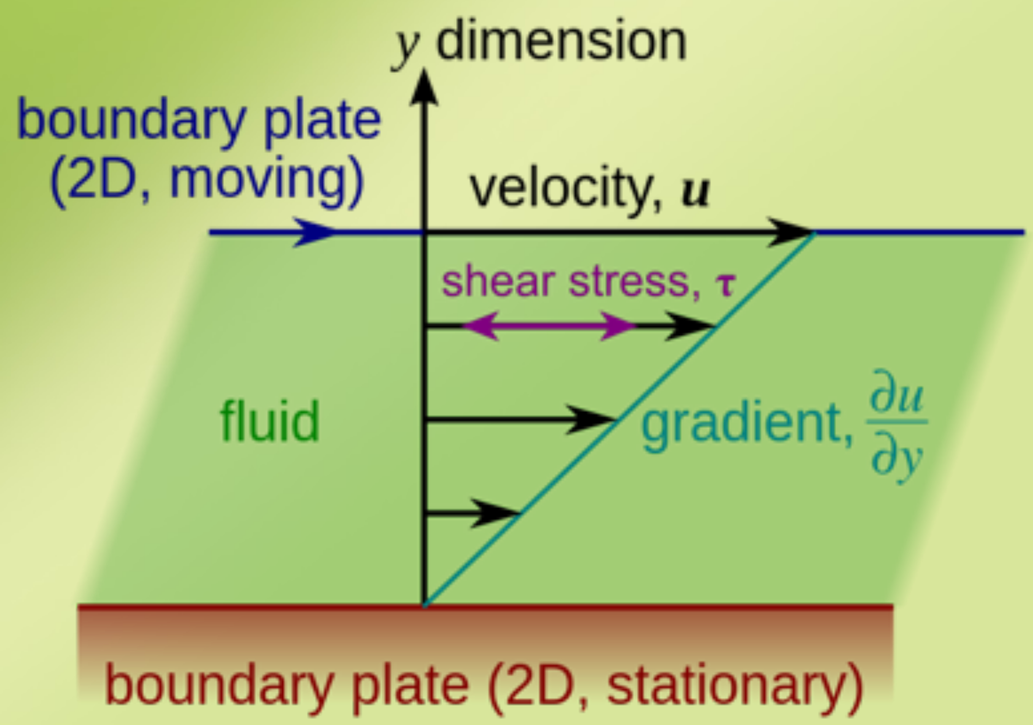
Adding new models

Create a hybrid system (a “testbed”) to examine key aspects of information mediation for linked (concurrent) multiscale simulations

- information transfer between models
- boundaries between models
- convergence of the solution
- stability
- …

Goal

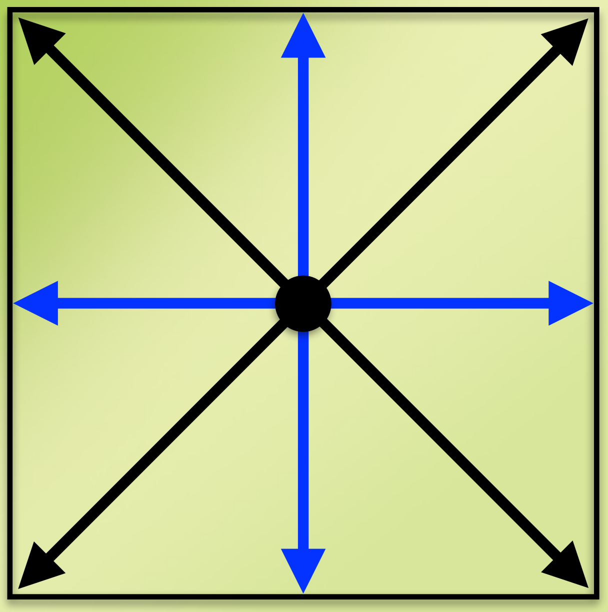
- fluid flow is calculated with the Lattice Boltzmann method
- interaction with surface modeled with molecular dynamics
- codes are coupled using the BMI



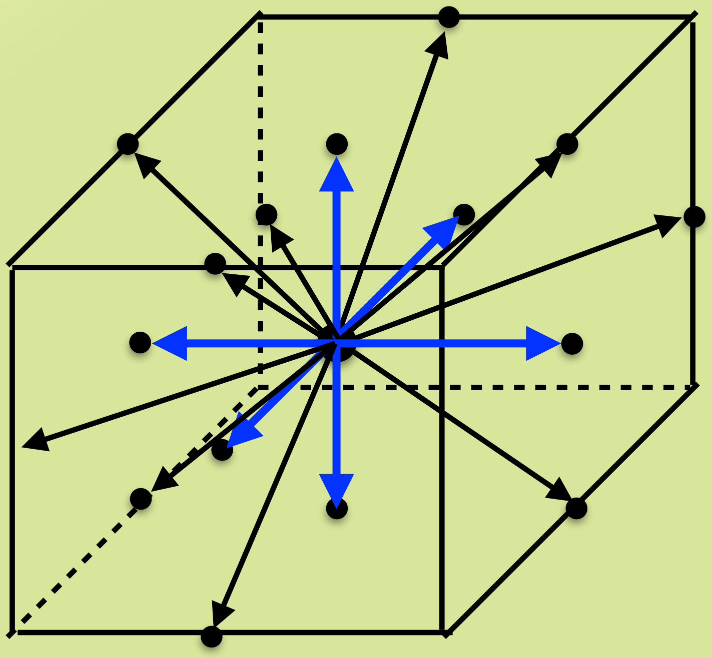
http://en.wikipedia.org/wiki/File:Laminar_shear.svg

Couette flow with surface interactions

Lattice Boltzmann method



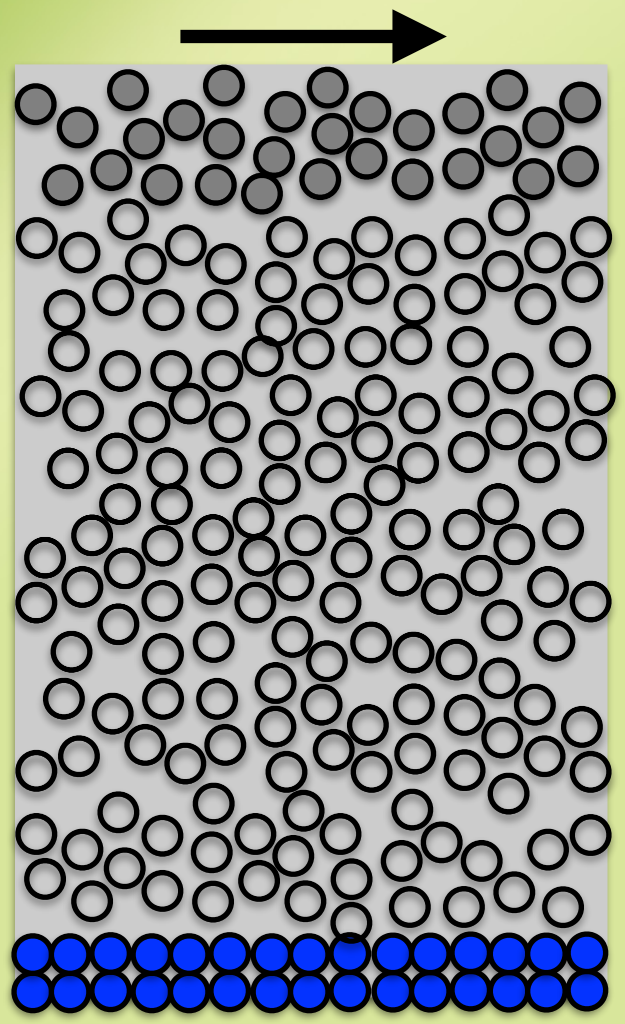
- a cellular automata
- single-particle distribution functions move along lattice sites
- “collisions” lead to equilibration
- measurable quantities: fluid density and velocity
- a lattice-based solution to Navier-Stokes equation



Chen and Doolen, Annu. Rev. Fluid Mech. 30, 329 (1998)

- calculate forces on the atoms from known interatomic potentials
- in this case, we used a Lennard-Jones potential
- solve Newton's equations using discrete time steps.
- time steps are small:

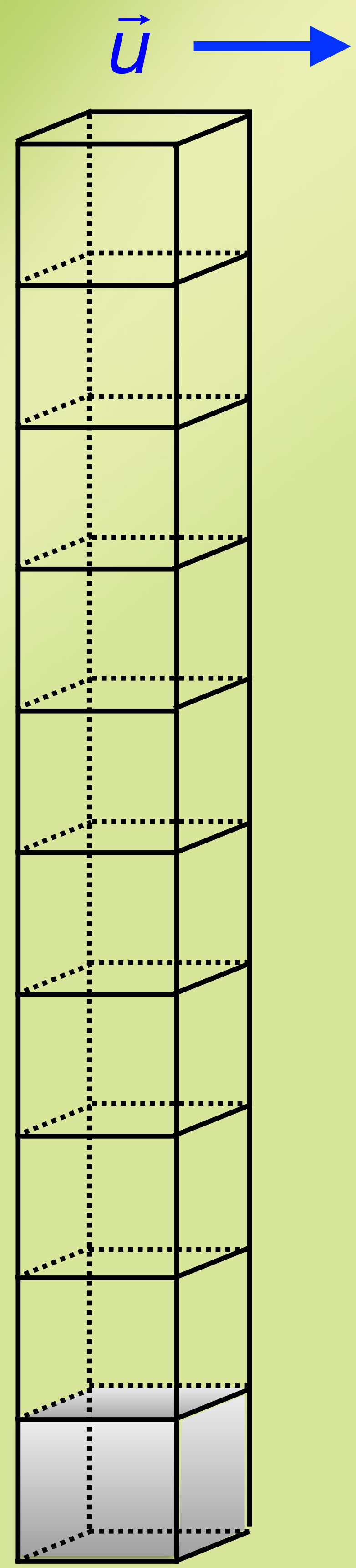
$$\Delta t \sim 10^{-14} - 10^{-15} \text{ s}$$



e.g., LeSar, Introduction to Computational Materials Science (Cambridge, 2013)

Molecular dynamics

- Lattice Boltzmann (LB) is used to calculate fluid flow on the full 3D grid
- molecular dynamics (MD) is used in the bottom grid volume to model atomic motions
- LB velocity sets boundary condition at top of molecular dynamics cell
- MD value for the slip velocity sets bottom boundary for LB



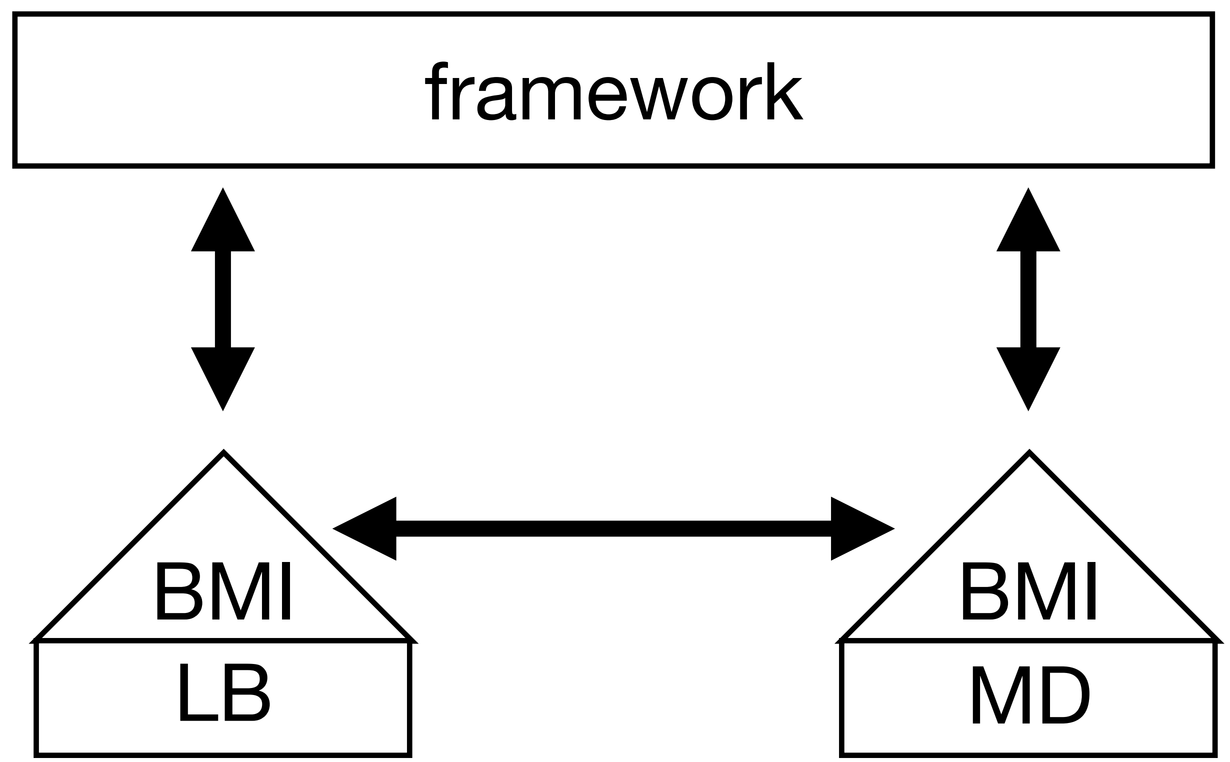
Boundary between LB/MD

- Each model has its own internal units.
- Each model is solved with its own time step (very different in size).
- Each model has its own implementation of boundary conditions.
- Each model has its own requirements for convergence.

The BMI represents information internal to the model through a standard set of functions.

void initialize(string input file, string identifier)	allocates memory for model and sets input variables
void run(int time steps)	runs model for number of time steps based on value of time steps
void finalize()	deallocates memory for model and prints output to a file
vector¡string¿ get_input_var_names()	returns list of input variables
vector¡string¿ get_output_var_names()	returns list of output variables
vector<string>get_boundary_condition_names()	returns list of usable boundary conditions
vector<string>get_boundary_condition_var_names(string boundary condition)	returns list of variables to use to enforce given boundary condition
string get var_type(string variable)	returns type of variable
string get_var_units(string variable)	returns units of variable
int get_var_rank(string variable)	returns rank of variable
double get_0d_double(string)	returns value of a zeroth rank floating point variable
vector<double>get_1d_double(string)	returns a first rank floating point variable
void set_2d_double_at_index(string, double, int, int)	set the value of a second rank floating point variable at a specified index
int get_3d_int_at_index(string, int, int, int)	return the value of a third rank integer variable at a specified index
vector<vector<vector<vector<string>>>>get_4d_string(string)	return a fourth rank string variable
void match_units(model *)	matches the values of variables in two models to put them into the same state
get and set functions exist for variables of every combination of data type and rank	

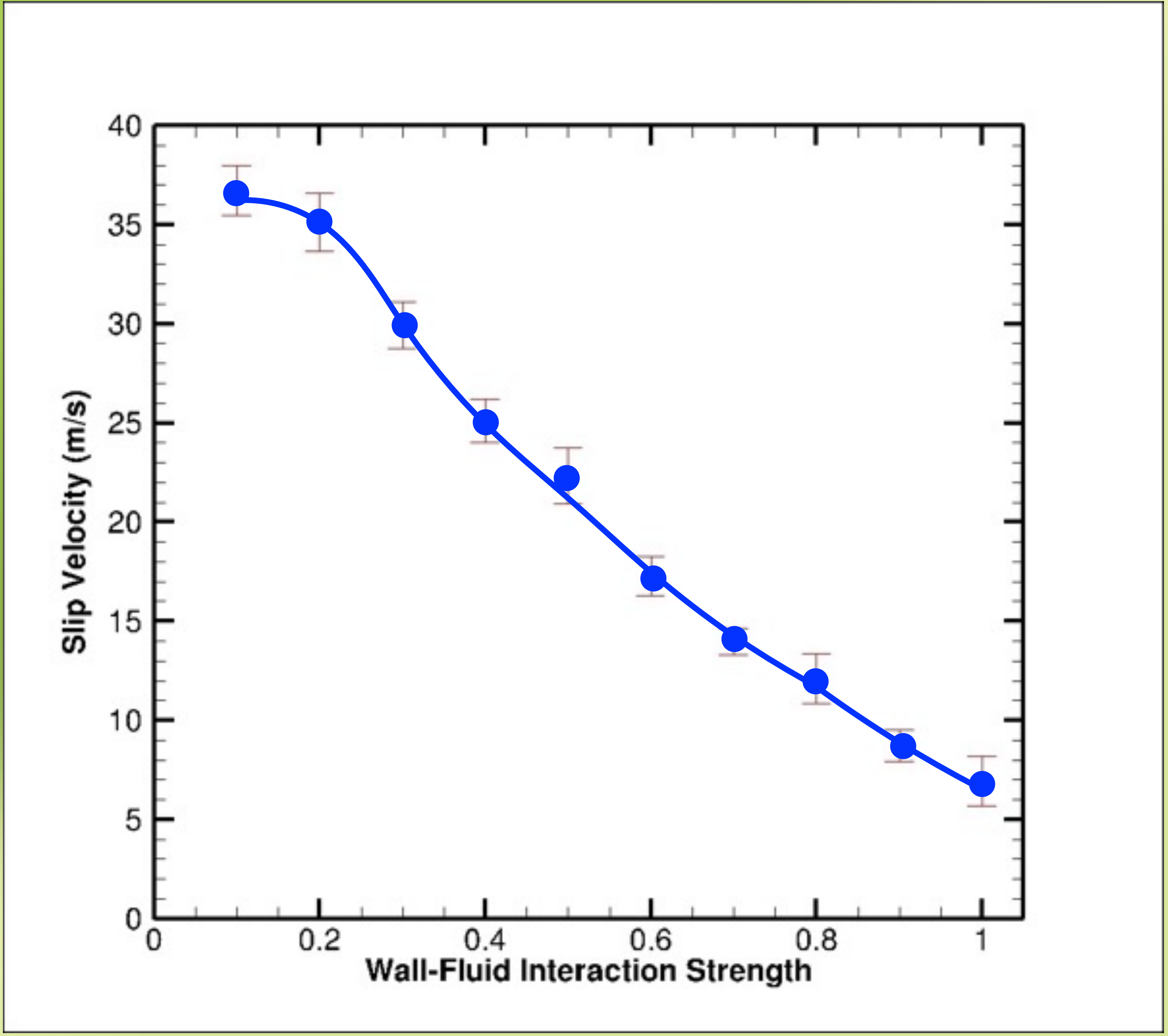
BMI functions



First application with BMI: 2 autonomous models

Our current focus is on understanding the boundary between methods linked with the BMI

- convergence
- stability
- boundary conditions



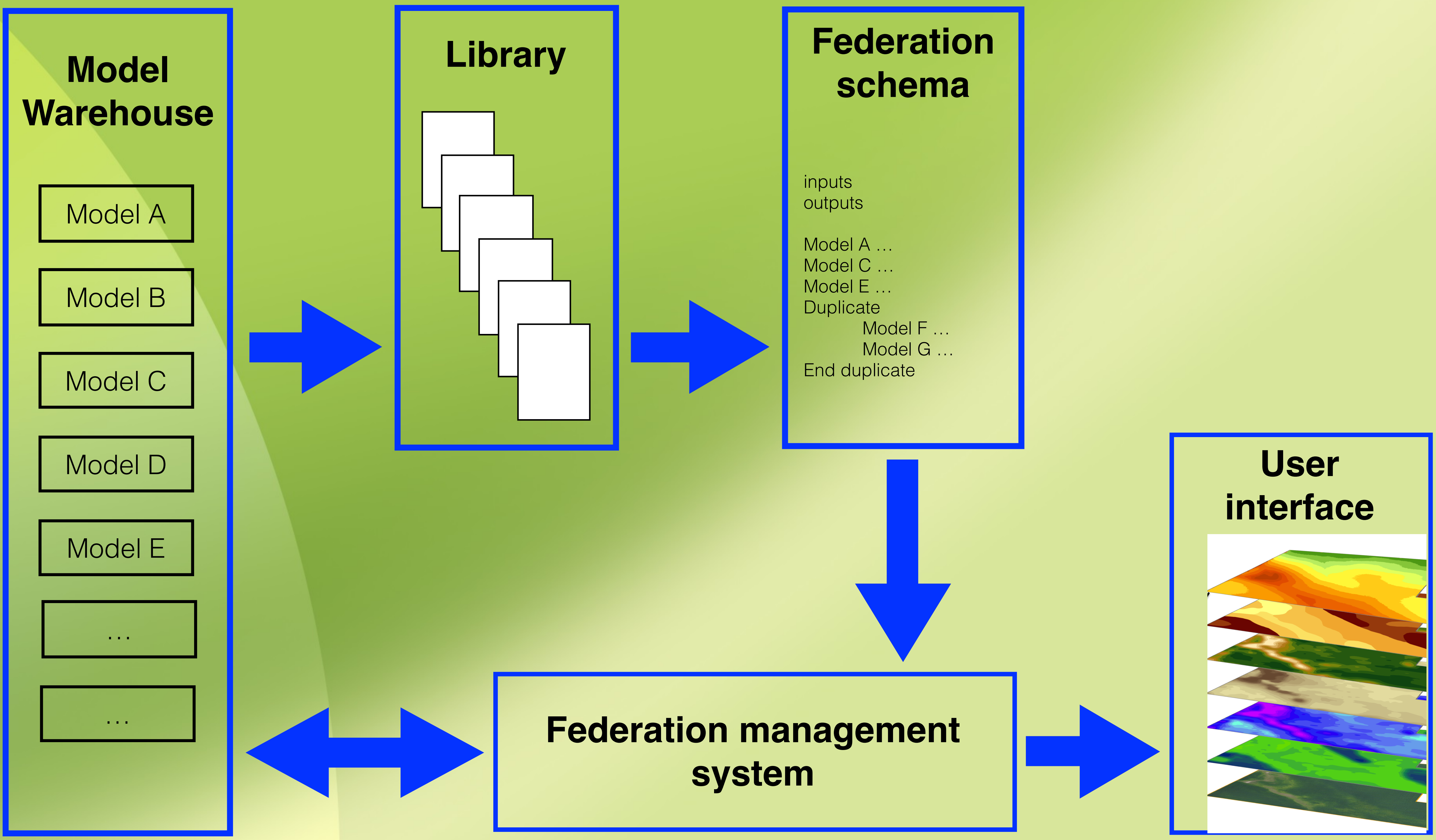
Results with BMI-linked code

- create or find independent models
- refine language for model-model communication
  - compatibility of information
  - boundaries
  - convergence
  - stability

  } mediation
- "snap" models together
  - substitute different models (e.g., models for solidification)

Path forward

To develop a library of materials models, enabling the DOE to create dynamic simulation tools in support of affordable, low carbon, high efficiency, advanced power systems.



Components and information flow

Richard LeSar

515-294-1841

lesar@iastate.edu