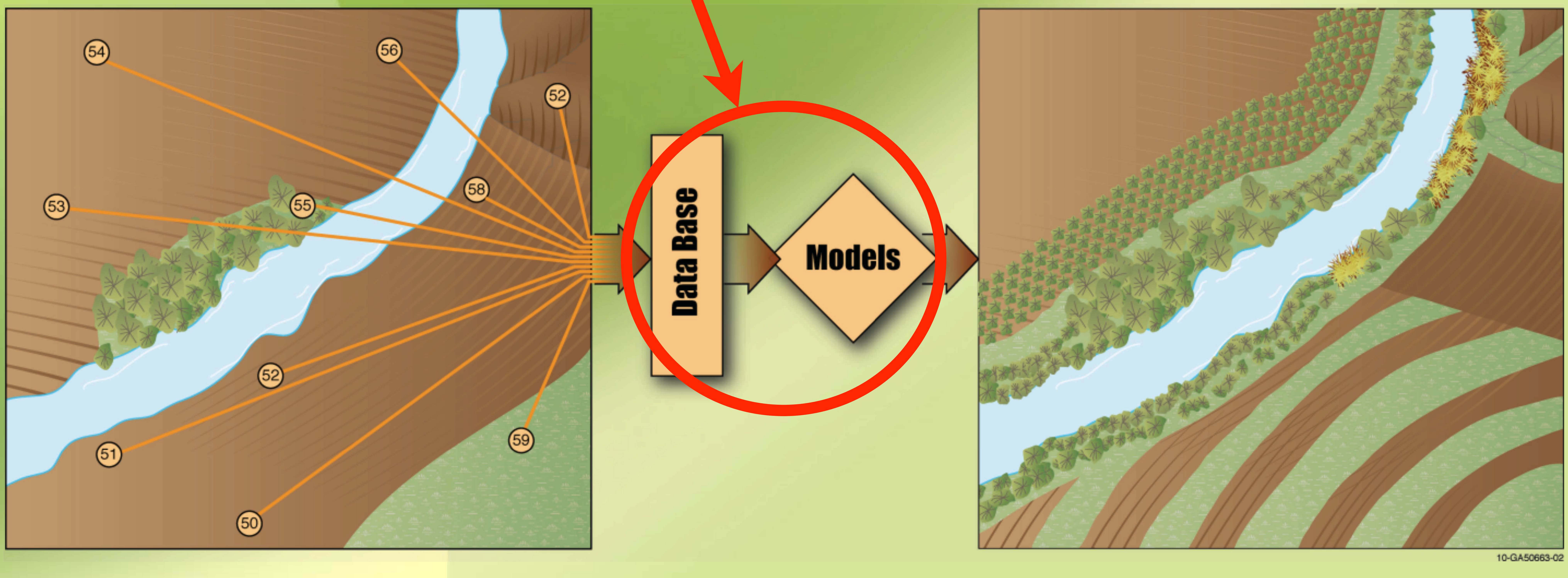
A Domain Specific Language to Simplify the Creation of Large Scale Federated Model Sets

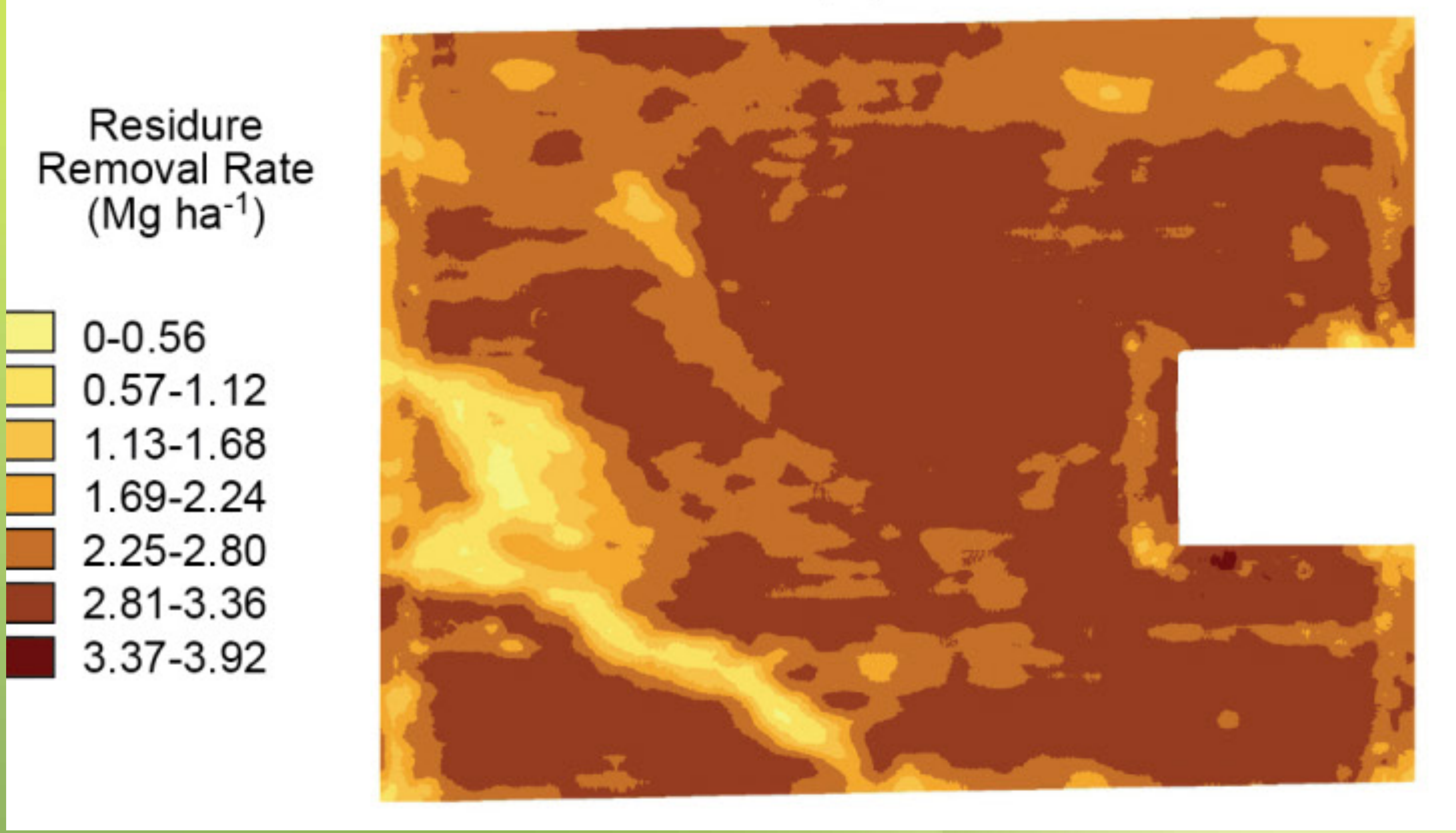
Zachary T. Reinhart
Sunil Suram
Kenneth "Mark" Bryden

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

- Cover crops
- Variable rate residue removal
- Integrated cropping systems
- Landscape management

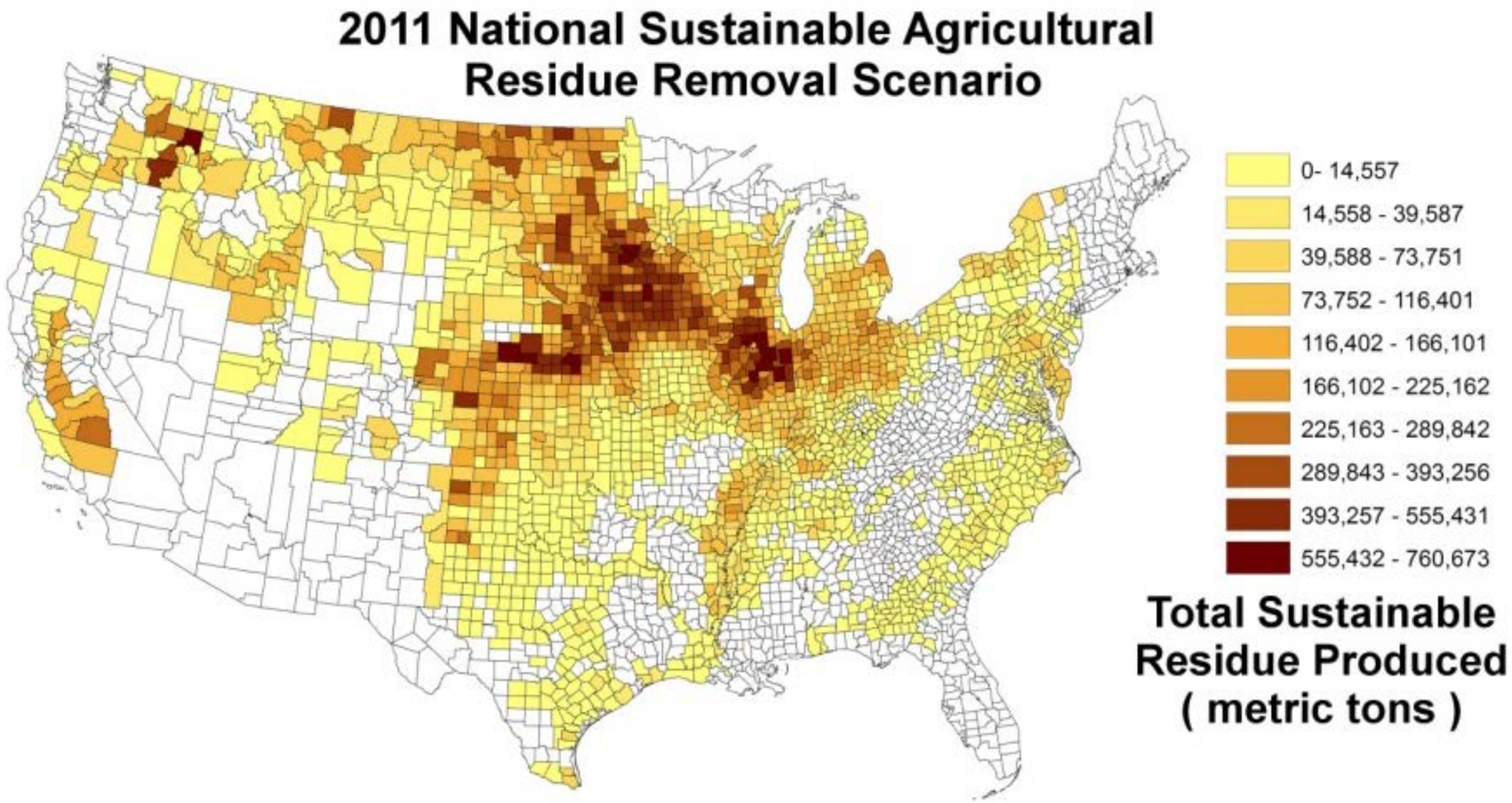


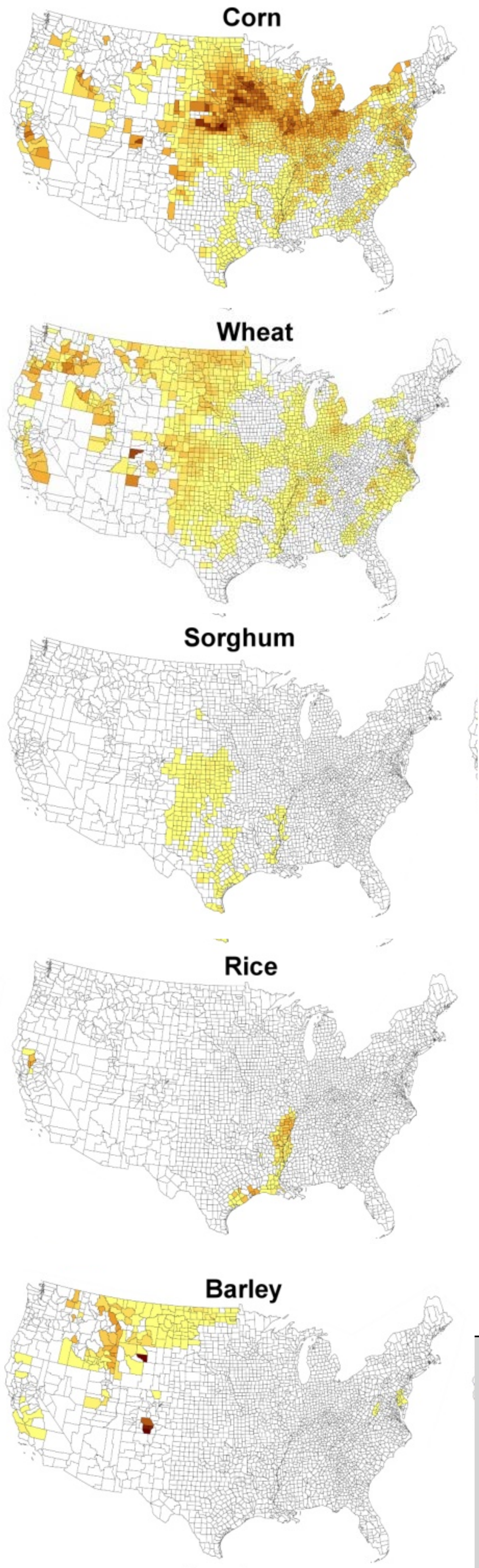
Agronomic strategies



Subfield analysis

DJ Muth, DS McCorkle, JB Koch, and KM Bryden, Agronomy Journal, 104(4):970-981 (2012)





Sustainable residue on a national scale

DJ Muth, KM Bryden, and RG Nelsen, Applied Energy, 102:403-417 (2013)

AGSOLVER

HOME | AGRONOMIC DECISION SERVICES | ABOUT AGSOLVER | CONTACT AGSOLVER

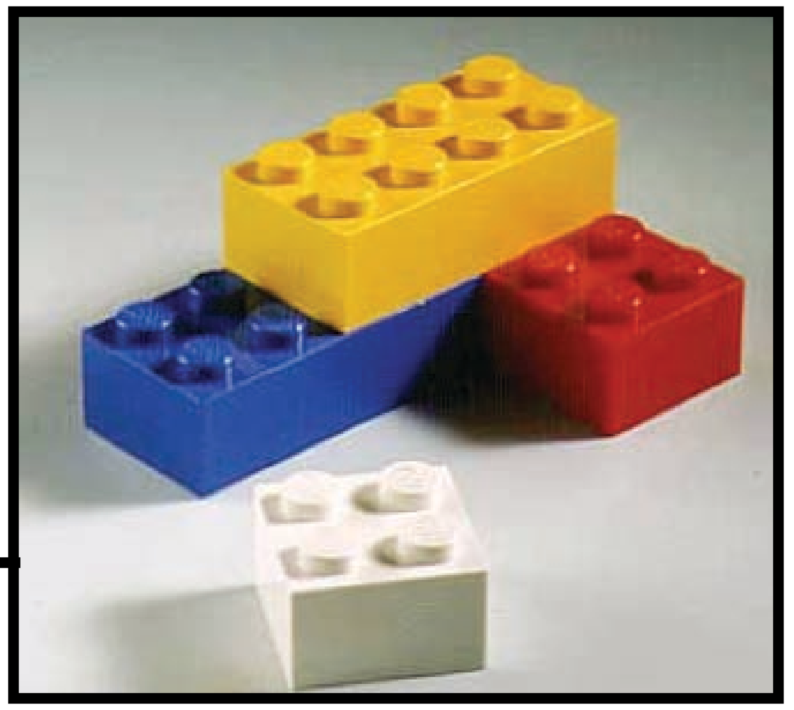
- Understand and Reduce Financial Risk
- Identify Opportunities to Increase Profitability
- Manage your Field at Subfield Scale
- Process Machine Data in any Format

AgSolver™

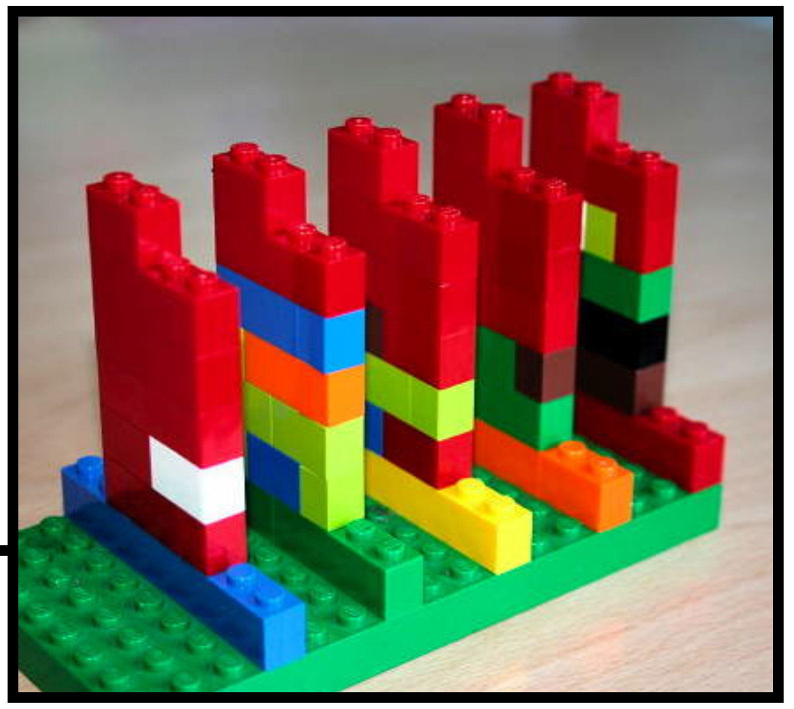
- Why can’t we integrate analysis into decision making on-the-fly?
- Why isn’t analysis and decision making like a game?
- Why do we continually make new models?

Three questions

Snap



Build

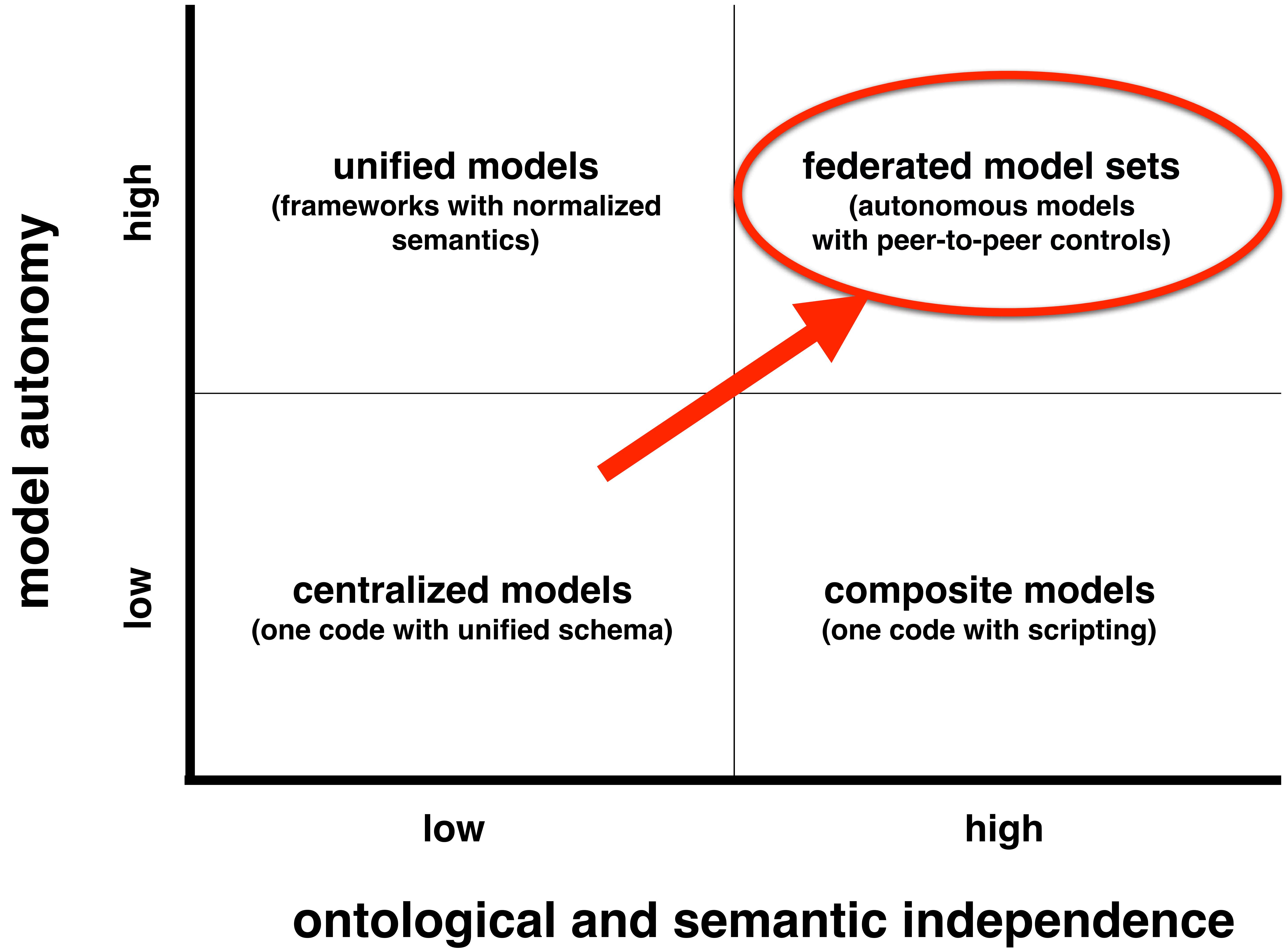


Do

New decision making paradigm

Requirements

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



Federated modeling

K. M. Bryden, 7th International Congress on Environmental Modelling and Software pp. 455-462 (2014)

Many different models

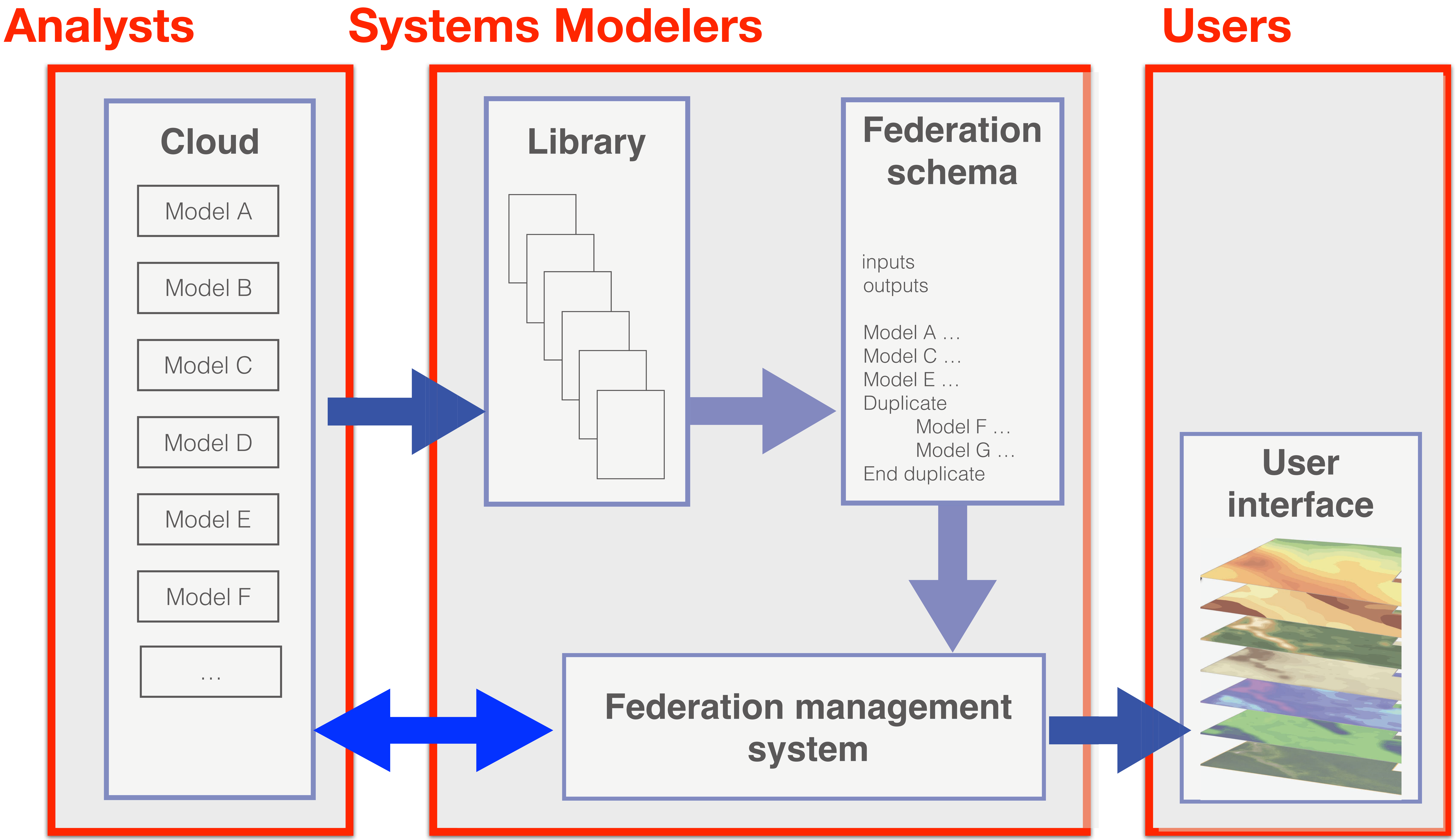
- No uniform, active storage space
- Not readily accessible, citable, or maintained
- Hard to locate and use existing code

Codes do not work together

- Systems models use codes specifically built for them
- Hard to use existing codes in a new systems model
- Clunky

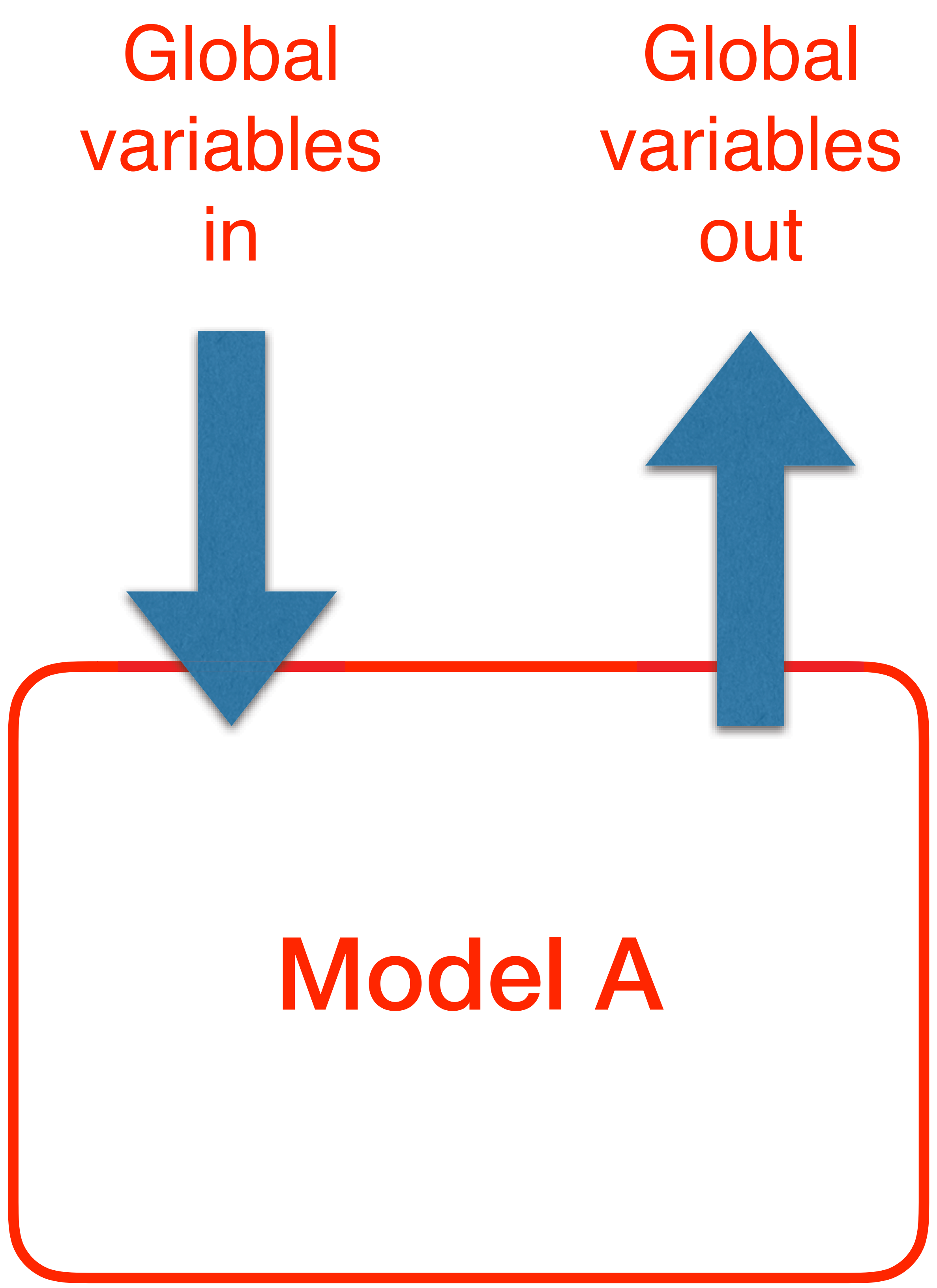
Systems modeling often lacks fidelity and granularity

- Algebraic expression, ODEs, reduced order models
- Averaging and message passing



Workflow

K. M. Bryden, 7th International Congress on Environmental Modelling and Software pp. 455-462 (2014)



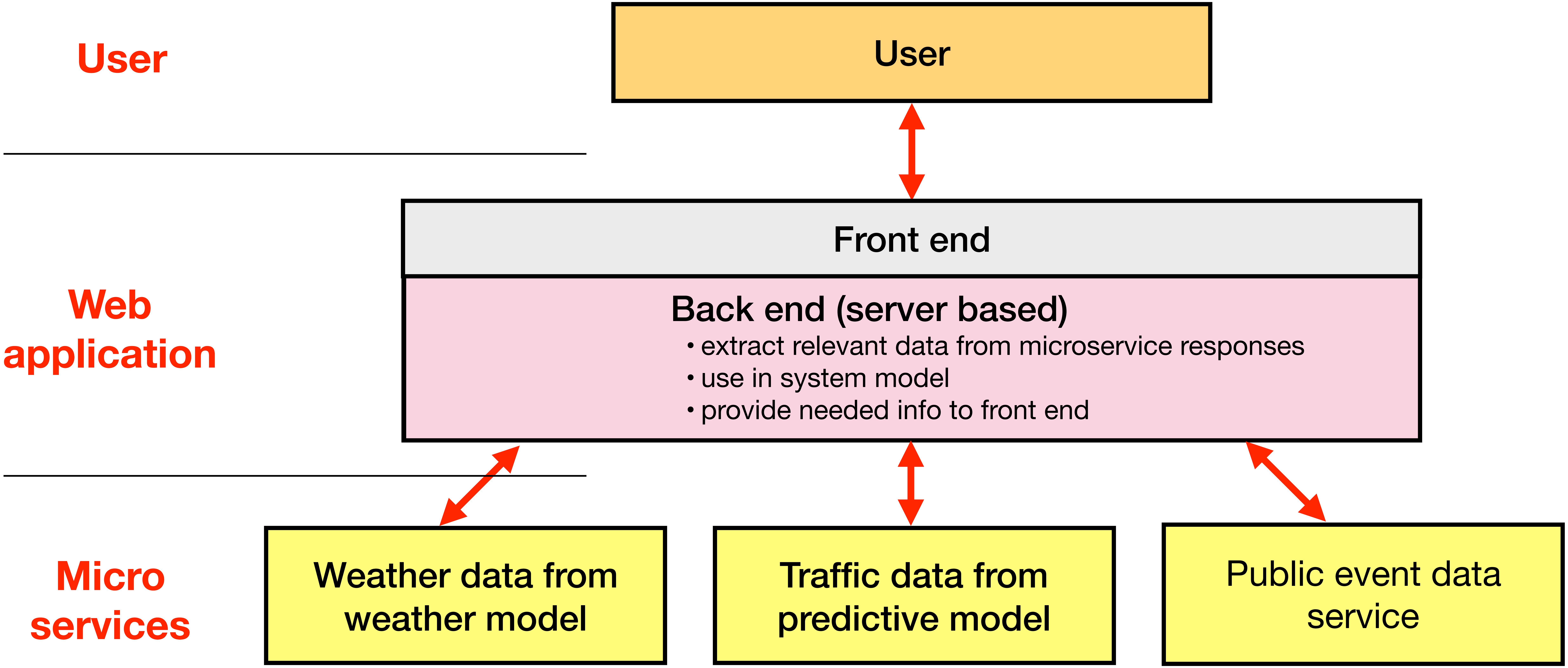
Each request is treated as an independent request

- communication is an independent set of a request and a response
- no session information is retained
- the code has no knowledge of the actions of other codes within the federation
- each member of the federation performs a specific task
- models are reusable for other analysis
- models can be strung together like beads on a complex weaving

Stateless modeling

- independently deployable
- easy assembly of various models and information sources
- models can be implemented using different programming languages, databases, hardware, and software environments
- direct replacement of a federation member can be performed without disruption to other members or the federation
- microservices perform (provide) one task only
- microservices are reusable

Properties of a microservices architecture



App implementation (microservices architecture)

Message contract system

A message contract consists of:

- A GUID
- A human-readable name
- A human-readable description
- A list of variables and types that make up the payload

Message Contract

Annual Hours of Fuel

GUID
e4c946

Data:

Name	Type
hours	double
type	string

Description:
Defines the annual hours used of a given fuel type.

Fuel Collection

Inputs

- Fraction of Collected Fuel
- Annual Energy Usage
- Fraction of Fuel Preparation
- Opportunity Cost of Fuel Collection

Outputs

Message Contract

e4c946

Description

Annual fuel collection time

Constraints

- Unit of time is hours.
- Numerical value must be positive.
- Fuel type must be one of known types.

Message Contract

Message Contract Number

e4c946

Data:

Name	Type
hours	double
type	string

Short title:

Annual Hours of Fuel

Description:

Defines the annual hours used of a given fuel type.

Fuel Cost

Inputs

Message Contract

e4c946

Description

Annual fuel collection time

Constraints

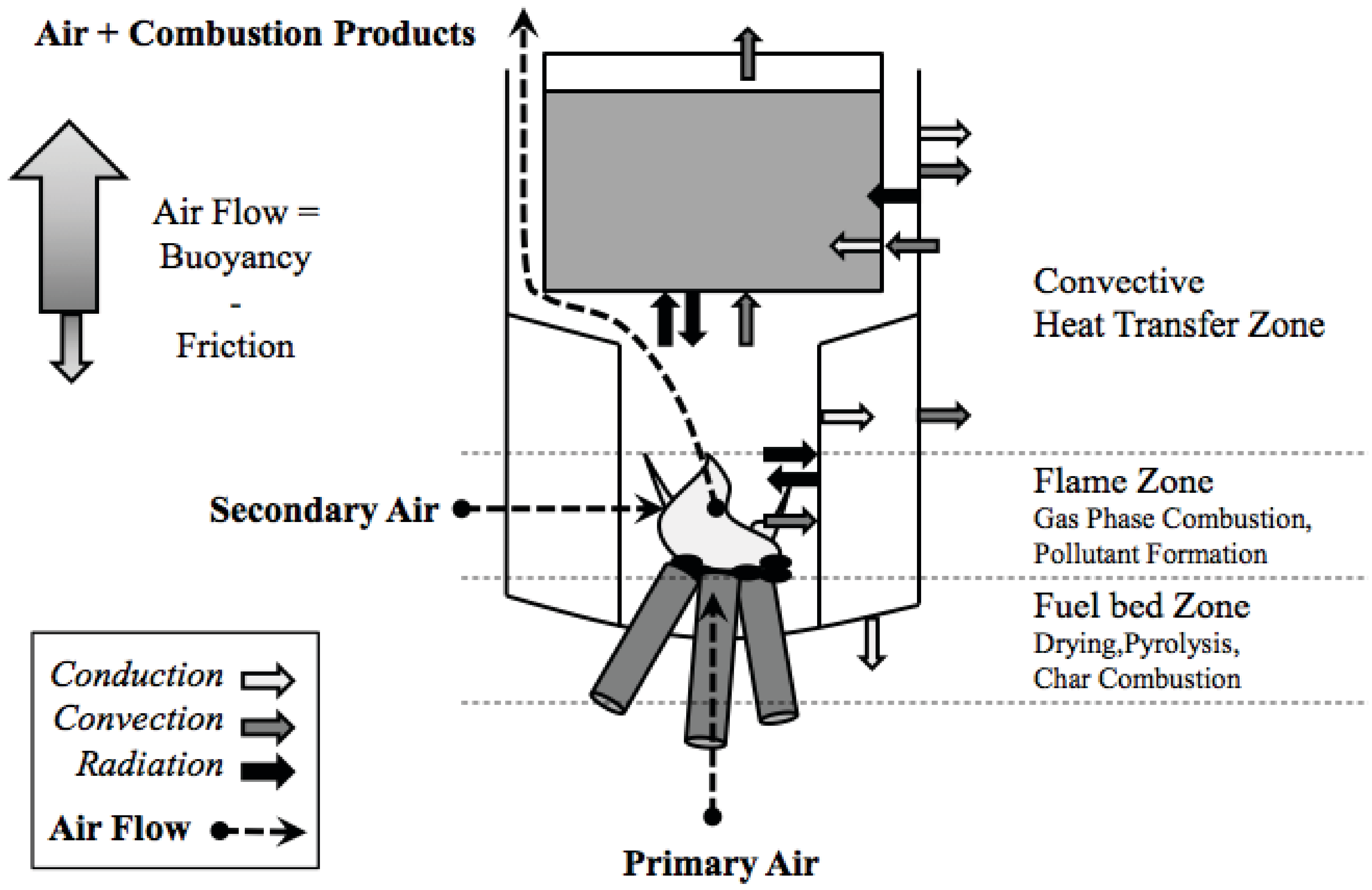
- Unit of time is hours.
- Numerical value must be positive.
- Fuel type must be one of known types.

- Fraction of Shadow Value of Time
- Cost of Human Labor

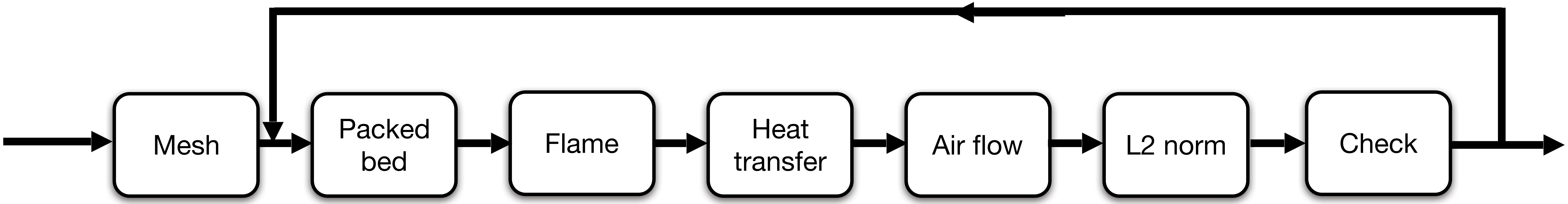
Outputs

- Annual Cost of Fuel Collection Time

Model coupling via message contract



- Based on existing monolithic cookstove model
- Re-implemented as seven stateless microservices
- Federation management system to manage model execution
- Internal communication via message queues
- System model accessible as web information service



Small cookstove model

NA MacCarty, KM Bryden, Energy for Sustainable Development, 33:96–107 (2016)

Model Name	Description
Mesh	Initializes geometry and allocates variables
Bed	Calculates rate of burning, production of fuel moisture and combustion products, and fuelbed and exit gas
Flame	Calculates rate of burning volatiles and exit gas temperature from the flame zone
Heat Transfer	Calculates exit temperatures through each control
Flow	Calculates velocity and pressure drop through each control volume in the flow path.
L2 norm	Calculates L2 norm between mass flow rates
Convergence	Sets a flag if the system of models has converged

Constituent models

- Running, proof-of-concept federated modeling system including
  - Model microservices
  - Federation management system
  - Message queue
  - Web front-end
- Simple web user interface
- Growing collection of component models: cookstoves, Hyper, …

But…

In order to build a federated model set, you must write code in Java.

Current toolset

Input blocks:

```
constants <system name>
 <constant 1>
 <constant 2>
 .
 .
 <constant n>
end constants
```

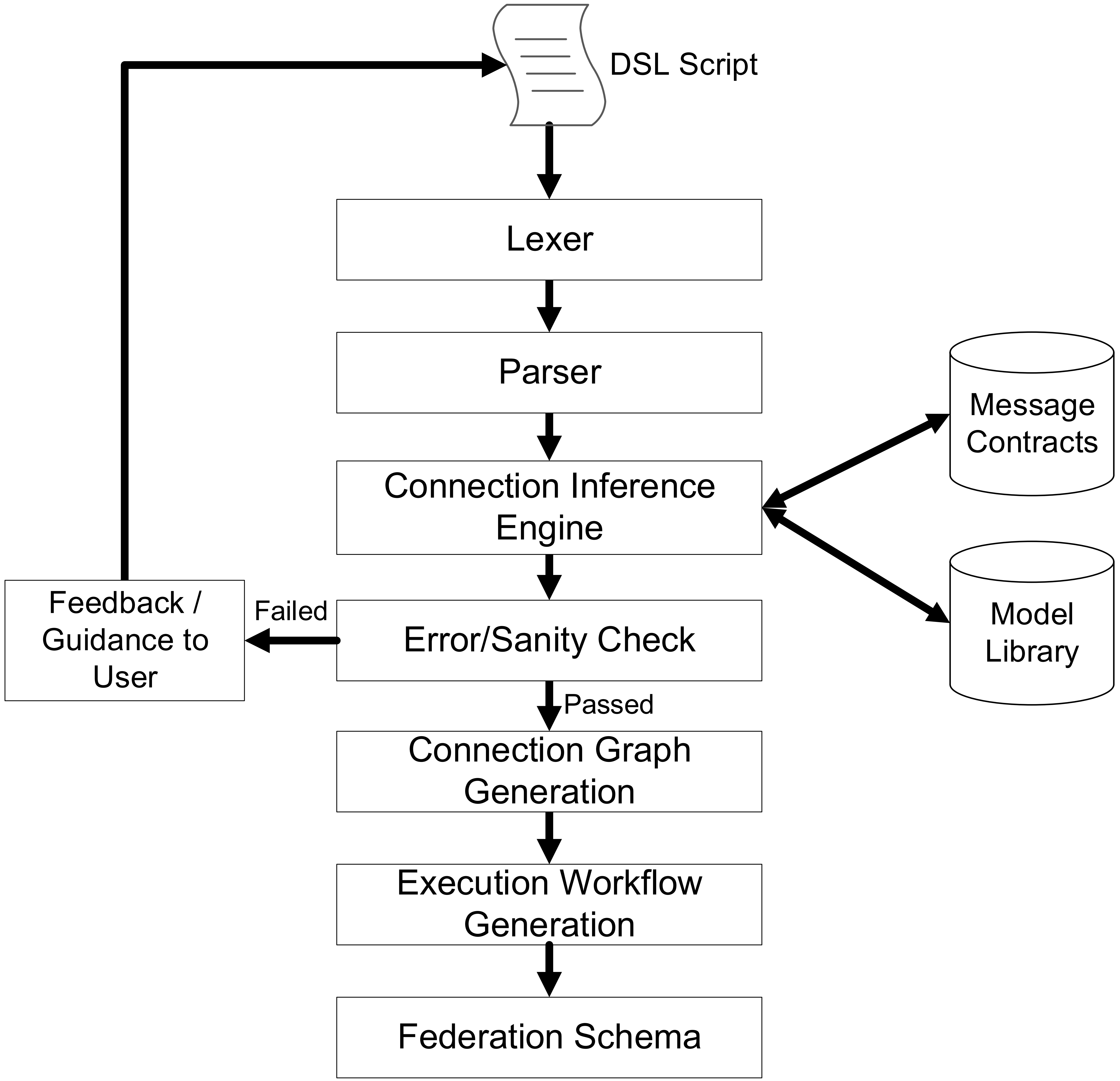
```
inputs <system name>
 <input 1>
 <input 2>
 .
 .
 <input n>
end inputs
```

Output block:

```
outputs <system name>
 <output 1>
 <output 2>
 .
 .
 <output n>
end outputs
```

System block:

```
system <system name> from library <library address>
 <model 1 name>
 <model 2 name>
 .
 .
 <model n name>
end system
```

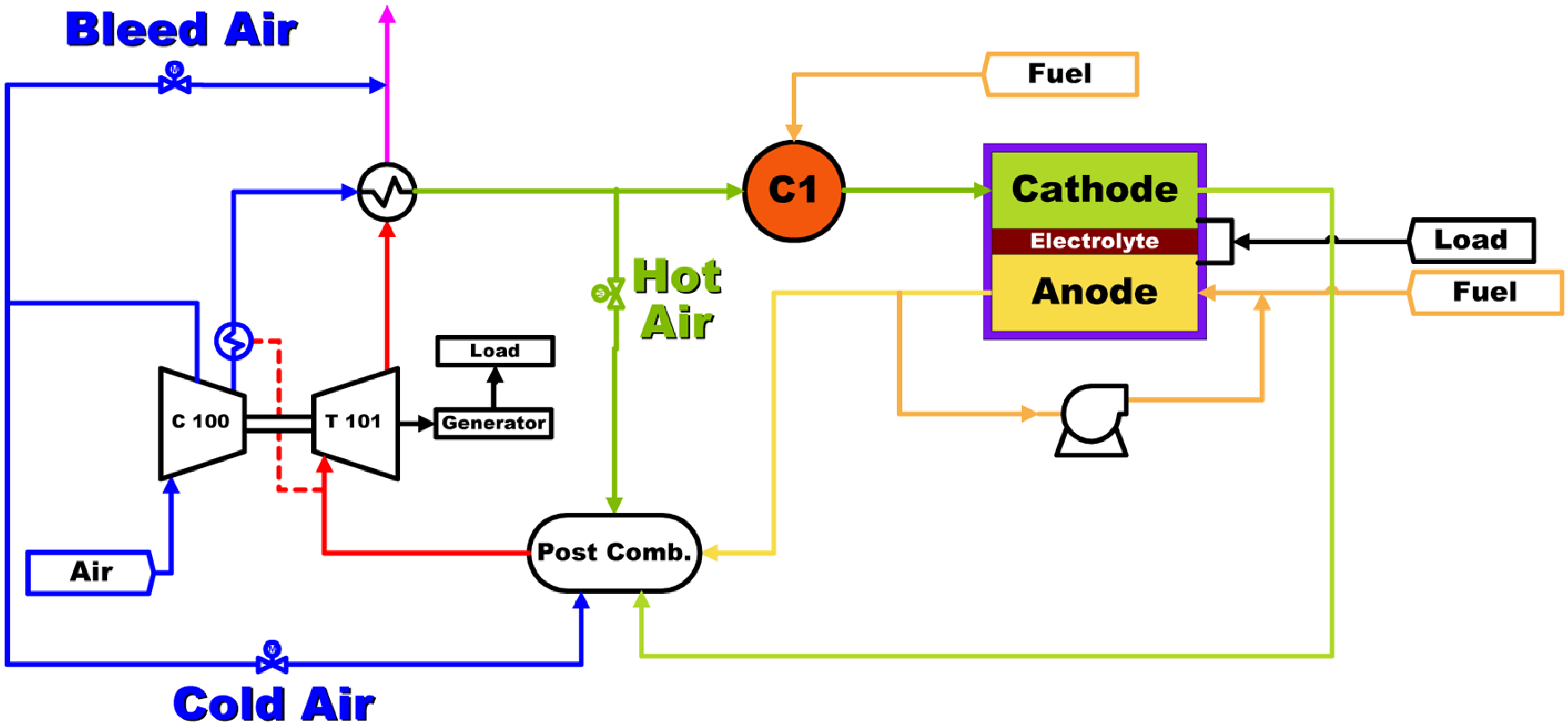


Domain specific language toolset

```
inputs stove_model
  H_sh
  W_sh
  W_pot
  W_c
  H_c

  D_c
  D_stove
  D_pot
  H_pot
end inputs

constants stove_model
  http://10.10.10.10/stovemodel
end constants
```

```
outputs stove_model
  stove_efficiency
end outputs

system stove_model :
  stove_mesh
  bed_model
  flame_model
  heat_model
  flow_model
  l2_norm
  convergence_check
end system
```

Implementation - Cookstove system



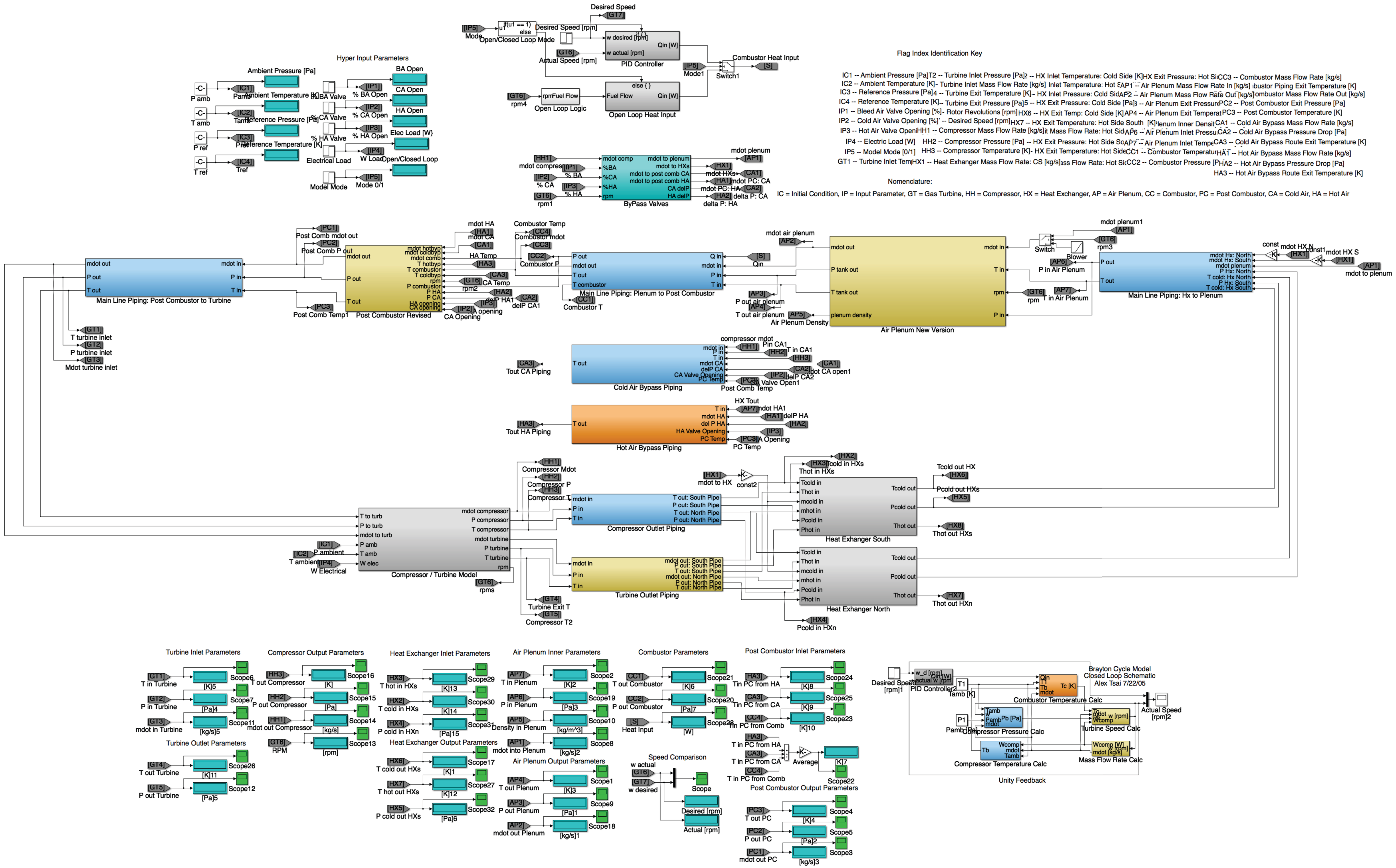
The hybrid performance project (Hyper)

Today

- A novel large scale cyber-physical platform for testing the dynamic performance of a hybrid fuel cell and gas turbine system
- ~1 MW
- Plan area = 50’ x 100’

Goal

- test the dynamic performance of any advanced power system that includes a gas turbine cycle



Hyper Simulink Model

A Tsai, PhD Thesis, West Virginia U, 2007

- Required the development of an object oriented support framework
- Each major component implemented as an independent class
- Components reused where feasible
- Common interface for each model
- Validated against original Simulink model
- 18,000 lines of code in 65 files

Hyper - high level systems

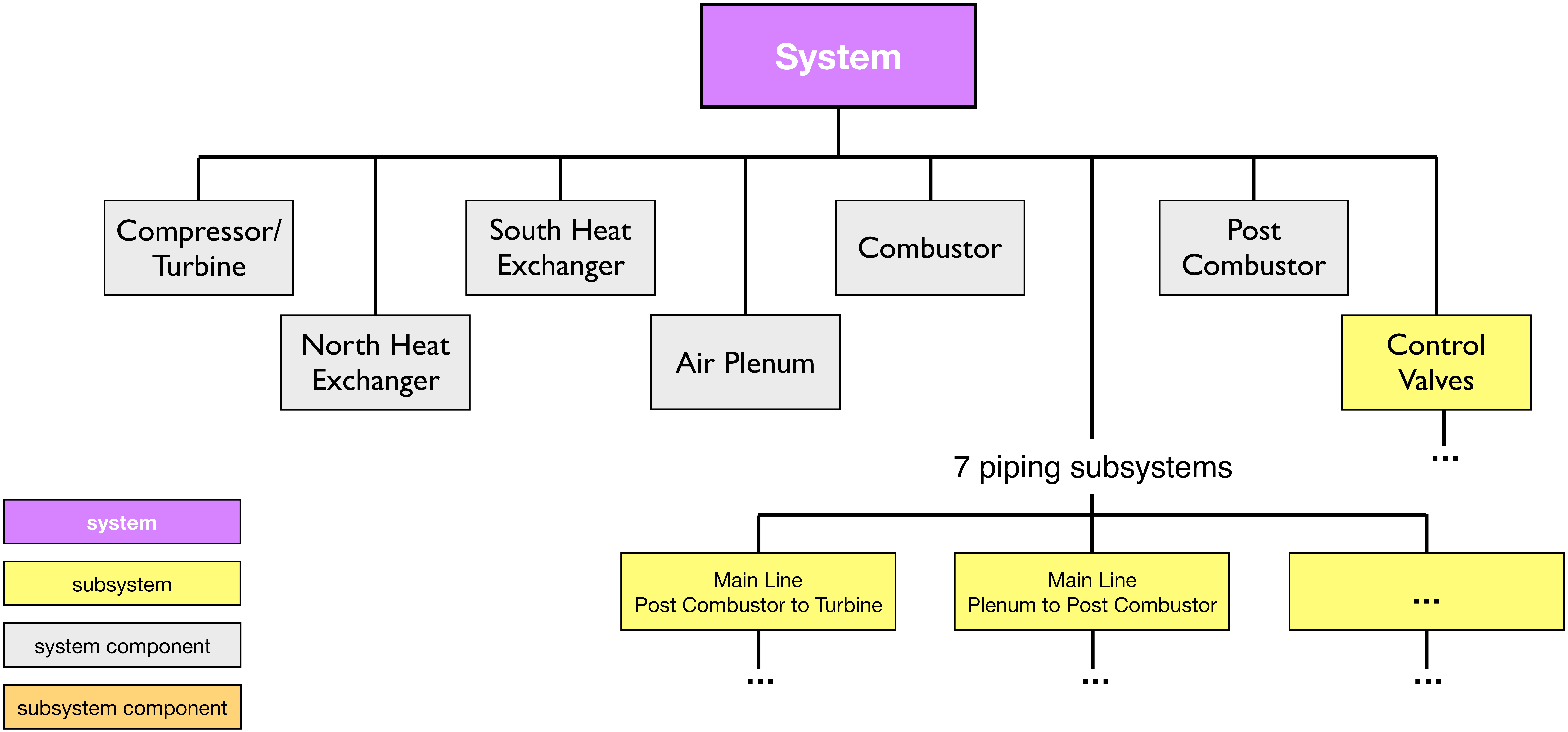
8 Microservices (models)

- Compressor/ Turbine
- Heat Exchanger
- Air Plenum
- Combustor
- Post Combustor
- Pipe Segment
- Pipe Fitting
- Control Valves

8 Message Contracts

- Temperature …
- Pressure …
- Mass Flow …
- Angular Frequency …
- Power …
- Pressure Loss …
- Density …
- Valve State …

Hyper — Microservices and message contracts



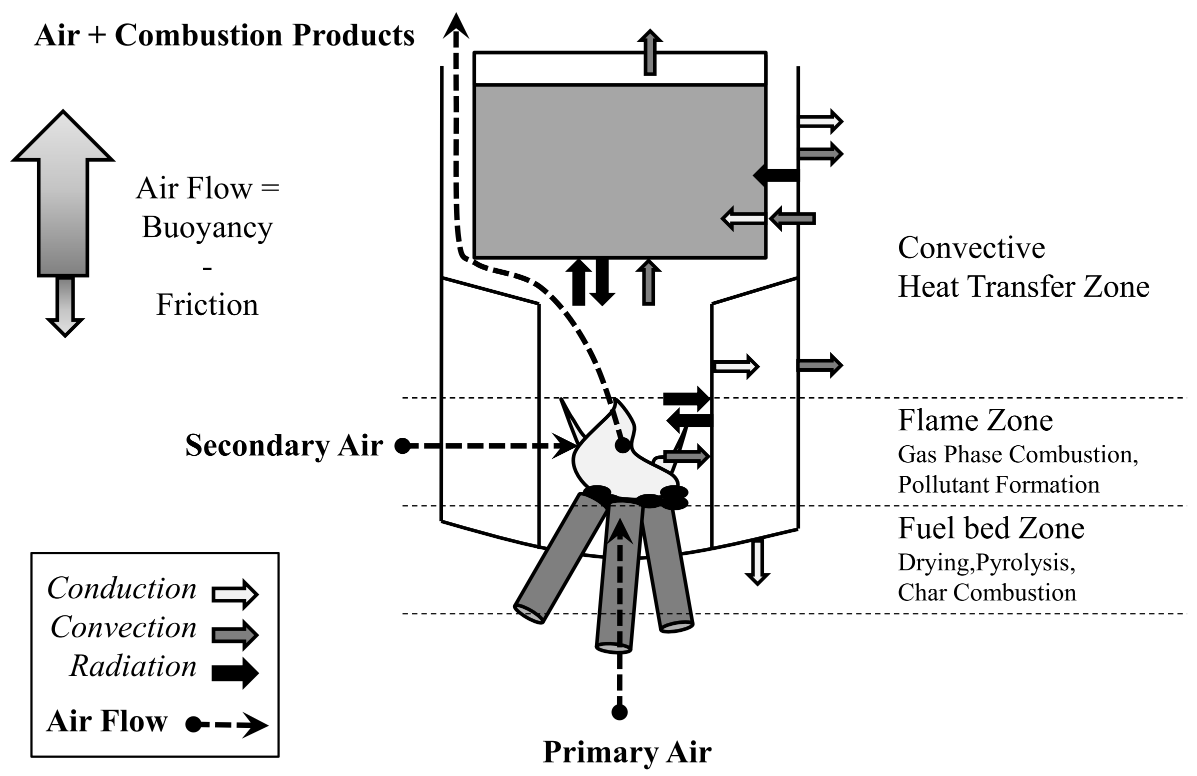
Hyper System of Models

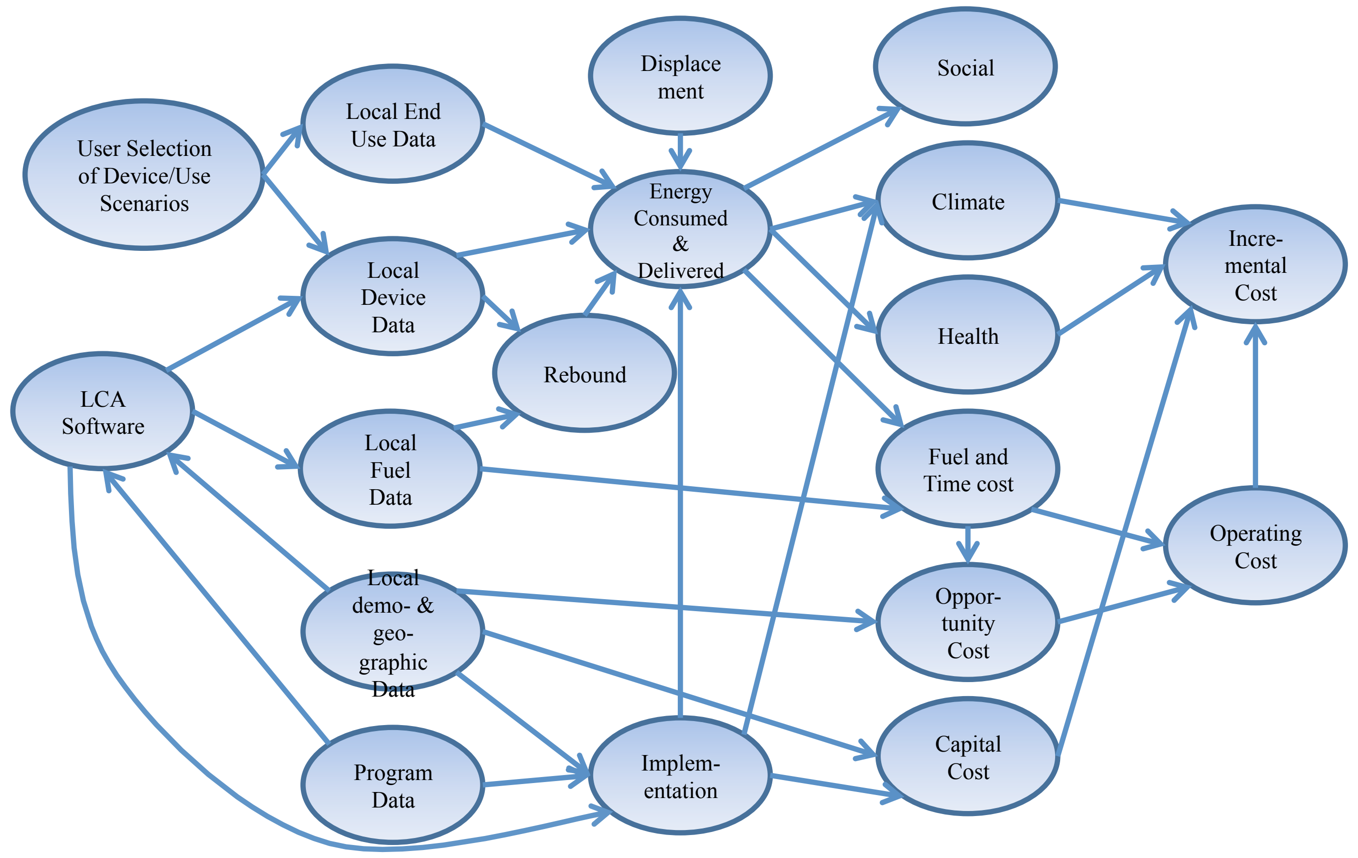
```
system hyper_model

  compressor_turbine
  air_plenum
  post_combustor
  heat_exchanger_north
  heat_exchanger_south
  combustor
  control_valves
  compressor_piping
  turbine_piping
  HX_plenum_piping
  plenum_postcomb_piping
  postcomb_turb_piping
  hot_bypass_piping
  cold_bypass_piping

  compressor_turbine -> compressor_piping
  compressor_turbine -> turbine_piping
  compressor_piping -> heat_exchanger_north
  compressor_piping -> heat_exchanger_south
  turbine_piping -> heat_exchanger_north
  turbine_piping -> heat_exchanger_south
  heat_exchanger_north -> HX_plenum_piping
  heat_exchanger_south -> HX_plenum_piping
  HX_plenum_piping -> air_plenum
  air_plenum -> plenum_postcomb_piping
  plenum_postcomb_piping -> post_combustor
  post_combustor -> postcomb_turb_piping
  postcomb_turb_piping -> compressor_turbine

end system
```

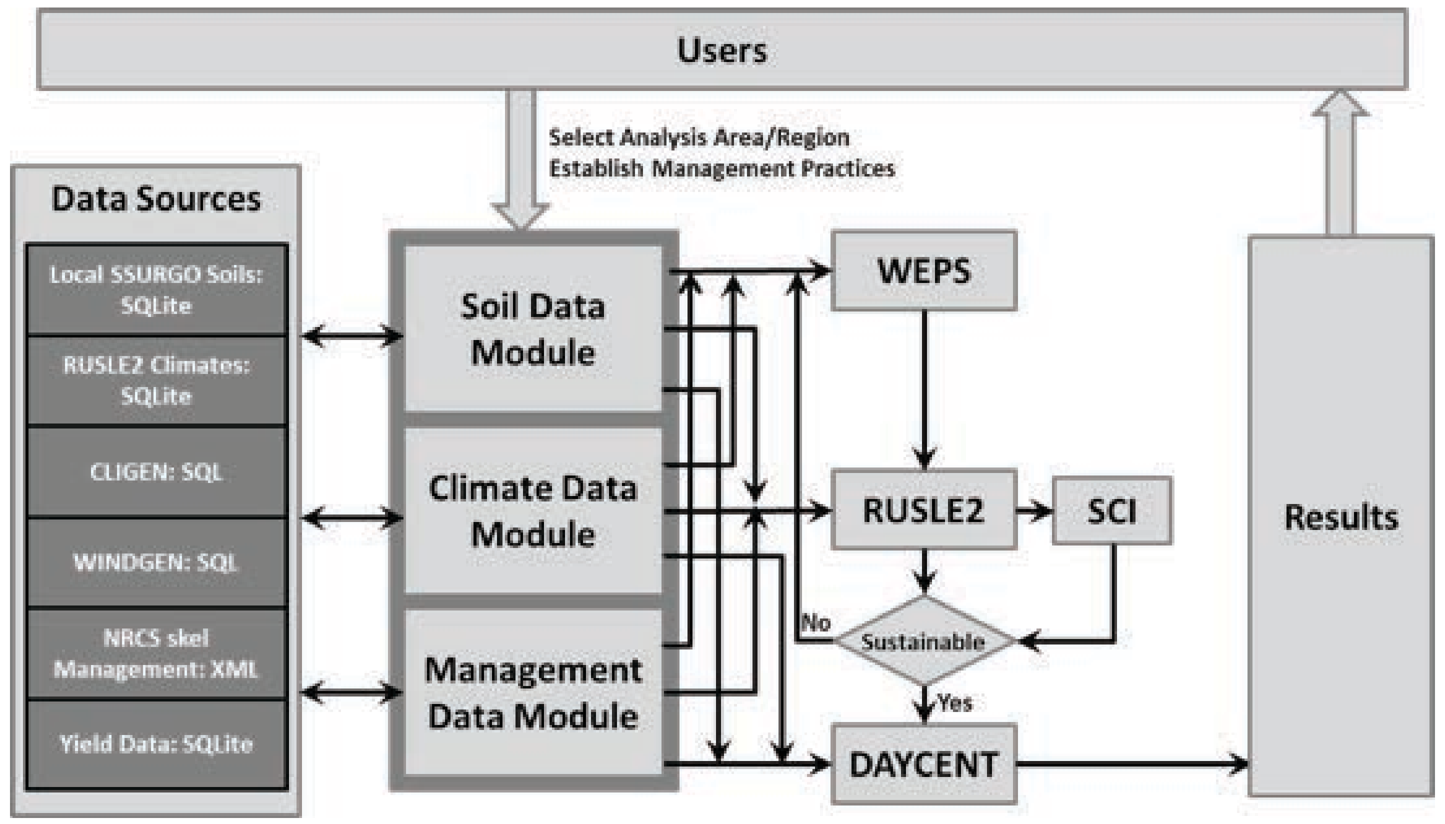
Hyper system DSL code

- What is the tradeoff between cost, acceptance, and impact for a particular geometry change to a cookstoves?
- How do I maximize the impact of my village energy system?
- If I change materials what will be impact of the stove on the local environment?

Decision based questions







Components

- Cookstoves
- Solar hot water
- Lights

+

Village Energy Model

- Rebound
- Climate impacts
- User acceptance

+

Agronomic Model

- Erosion
- Fertility
- Crop yield

Systems level design

nt and former PhD students

Tina Akinyi
Dan Bell
Steve Corns
Peter Finzell
Steven Gent
Gengxun Huang
Nate Johnson
Peter Johnson
Balu Karthikeyan
Nordica MacCarty
Doug McCorkle
David Muth
Zach Reinhart
Sunil Suram
Aditya Velivelli

Collaborators

Prof. Dan Ashlock
Dr. Aaron Bryden
Prof. Kris Bryden
Prof. Arne Hallam
Prof. Richard LeSar
Prof. Tom Shih

Major Funders

U.S. Department of Energy, Office of Fossil Energy
U.S. Department of Energy, Office of Energy Efficiency and Renewable Energy
U.S. Army, Armament Research, Development and Engineering Center
Deere & Co

And many undergraduate and masters student researchers and our colleagues and friends in the ISU Virtual Reality Applications Center

Acknowledgements

Professor Mark Bryden
+1 515-294-3891
kmbryden@iastate.edu

Contact info