



# ANS MultiApps tutorial

November 2022

*Changing the World's Energy Future*

Guillaume Louis Giudicelli, Cody J Permann, Derek R Gaston, E. Shemon, Y. Miao, A. Abdelhameed, F. Kong



*INL is a U.S. Department of Energy National Laboratory operated by Battelle Energy Alliance, LLC*

#### **DISCLAIMER**

This information was prepared as an account of work sponsored by an agency of the U.S. Government. Neither the U.S. Government nor any agency thereof, nor any of their employees, makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness, of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. References herein to any specific commercial product, process, or service by trade name, trade mark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the U.S. Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the U.S. Government or any agency thereof.

# **ANS MultiApps tutorial**

**Guillaume Louis Giudicelli, Cody J Permann, Derek R Gaston, E. Shemon, Y.  
Miao, A. Abdelhameed, F. Kong**

**November 2022**

**Idaho National Laboratory  
Idaho Falls, Idaho 83415**

**<http://www.inl.gov>**

**Prepared for the  
U.S. Department of Energy  
Under DOE Idaho Operations Office  
Contract DE-AC07-05ID14517**



NRIC

National  
Reactor  
Innovation  
Center

INL/CON-22-XXXXX  
ANS Winter 2022

# Assembling Multiphysics Nuclear Reactor Simulations Using the MOOSE Framework

E. Shemon, Y. Miao, A. Abdelhameed

Argonne National Laboratory

C. Permann, A. Lindsay, **G. Giudicelli**, F. Kong, D. Gaston

Idaho National Laboratory



# Mission Statement

**NRIC:** Deliver successful demonstration and deployment of advanced nuclear energy

- **EBR-II Test Bed (DOME)**
- **ZPPR Test Bed (LOTUS)**
- **Virtual Test Bed (VTB):** Accelerate deployment of advanced reactors by leveraging state-of-the-art Modeling and Simulation (ModSim) tools developed by the NEAMS program

Ok, but what is it?  
Reactor Demonstrations

## 1) Model Development

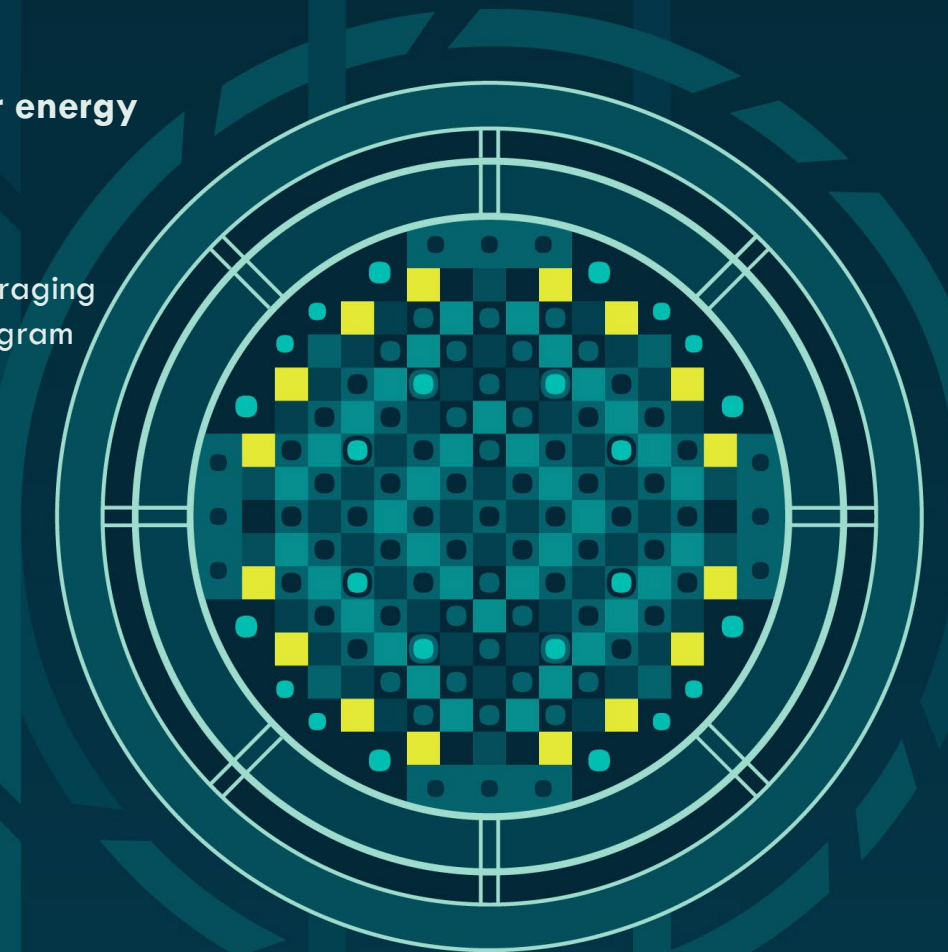
Leverage codes supported by NEAMS to provide powerful, highly-adaptable simulations:

- Analysis Tools
- Problem benchmarks
- Testing capabilities
- Safety review



## 2) Model Repository

- Storing and showcasing open use-cases
- Stewardship of industry-relevant models
- Integration with code development framework to avoid legacy issues



# The VTB Repository

[https://mooseframework.inl.gov/virtual\\_test\\_bed](https://mooseframework.inl.gov/virtual_test_bed)

## 2. GitHub Repo

Ability to clone and contribute to models

vtb

ContributingCitingGitHub

### Virtual Test Bed

Welcome to the Virtual Test Bed (VTB) repository!

This repo is a National Reactor Innovation Center (NRIC) initiative aiming to facilitate the use of advanced modeling & simulation (M&S) tools developed by the Department of Energy NEAMS program. Following the passage of the Nuclear Energy Innovation Capabilities Act, NRIC was created for accelerating the deployment of these novel reactor concepts. This will be achieved by providing both physical and virtual spaces for building and testing various components, systems, and complete pilot plants. The VTB represents the virtual arm in collaboration with DOE's NEAMS program.

The VTB repo hosts a wide variety of example challenge problems based on advanced reactor designs. This website contains the background and documentation for each use case. Click on a reactor type below to find out more. To access the corresponding inputs and supporting file, refer to the [repository](#).

Users will need to request access to the controlled NEAMS software from the Nuclear Computational Resource Center. For additional information on the VTB, please reach out to Dr. Abou-Jaoude at [abdalla.aboujaoude@inl.gov](mailto:abdalla.aboujaoude@inl.gov).

#### Information about the Virtual Test Bed


- How to use the Virtual Test Bed?
- Multiphysics reactor modeling using the MultiApps system
- Frequently Asked Questions and Discussion Forum
- Contribution guidelines to add a new model

#### Reactor Models

- Models sorted by type of simulation
- Molten Salt Reactor (MSR)
- High Temperature Gas-Cooled Reactor (HTGR)
- Pebble Bed Fluoride-Salt-Cooled High-Temperature Reactor (PB-FHR)
- Sodium Fast Reactor (SFR)
- Heat-Pipe Micro Reactor (MPR)

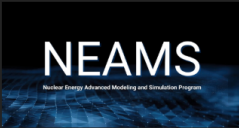
NOTE

Link to the GitHub repository



NRIC

National Reactor Innovation Center



TestsActionsProjectsWikiSecurityInsights

main2 branches0 tagsGo to fileCode

GludGlud Use public links in repository, refs #34614981 hour ago88 commits

doc

Use public links in repository, refs #31 hour ago

htg/mhtr

Limit each test to the relevant applications that can run them, refs #82 days ago

msfr

Limit each test to the relevant applications that can run them, refs #82 days ago

pbfr/reflector

Limit each test to the relevant applications that can run them, refs #82 days ago

sfr

Move SFR assembly model23 hours ago

.gitattributes

Fixup for ifs files21 hours ago

.gitignore

Add FHR steady state input files2 days ago

COPYRIGHT

Fixup for ifs files21 hours ago

LICENSE

Add license, copyright and readme to repository2 days ago

README.md

Use public links in repository, refs #31 hour ago

About

The National Reactor Innovation Center's (NRIC) Virtual Test Bed Repository

ReadmeView license

Releases

No releases published

Packages

No packages published

Contributors

GludGlud Guillaume Giudicelli

milijm Jason Miller

Languages

Assembly 59.9%SWIG 18.8%

Fortran 14.2%Python 6.9%

E 0.2%

README.md

Welcome to the Virtual Test Bed (VTB) repository!

This repo is a National Reactor Innovation Center (NRIC) initiative aiming to facilitate the use of advanced modeling & simulation (M&S) tools developed by the Department of Energy NEAMS program. Following the passage of the Nuclear Energy Innovation Capabilities Act, NRIC was created for accelerating the deployment of these novel reactor concepts. This will be achieved by providing both physical and virtual spaces for building and testing various components, systems, and complete pilot plants. The VTB represents the virtual arm in collaboration with DOE's NEAMS program.

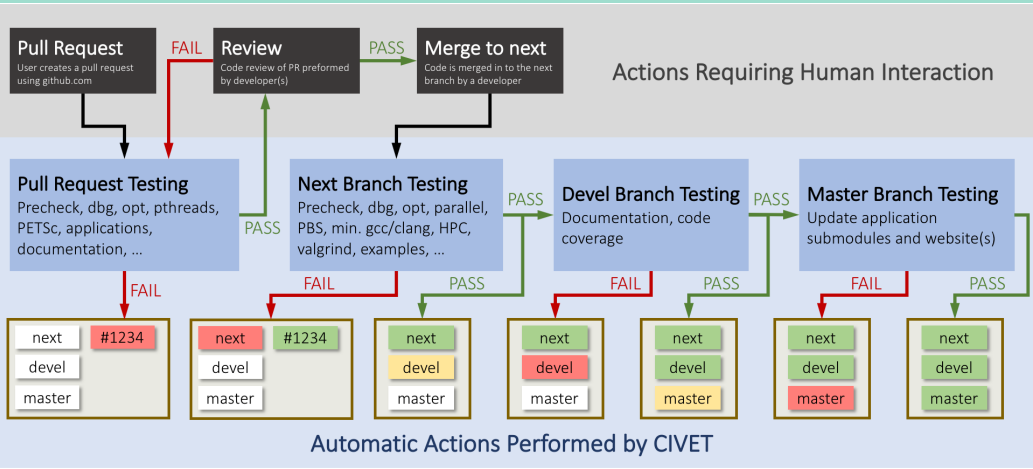
The VTB repository hosts a wide variety of example challenge problems based on advanced reactor designs. Please refer to the documentation website for background and documentation for each use case.

## 1. Documentation

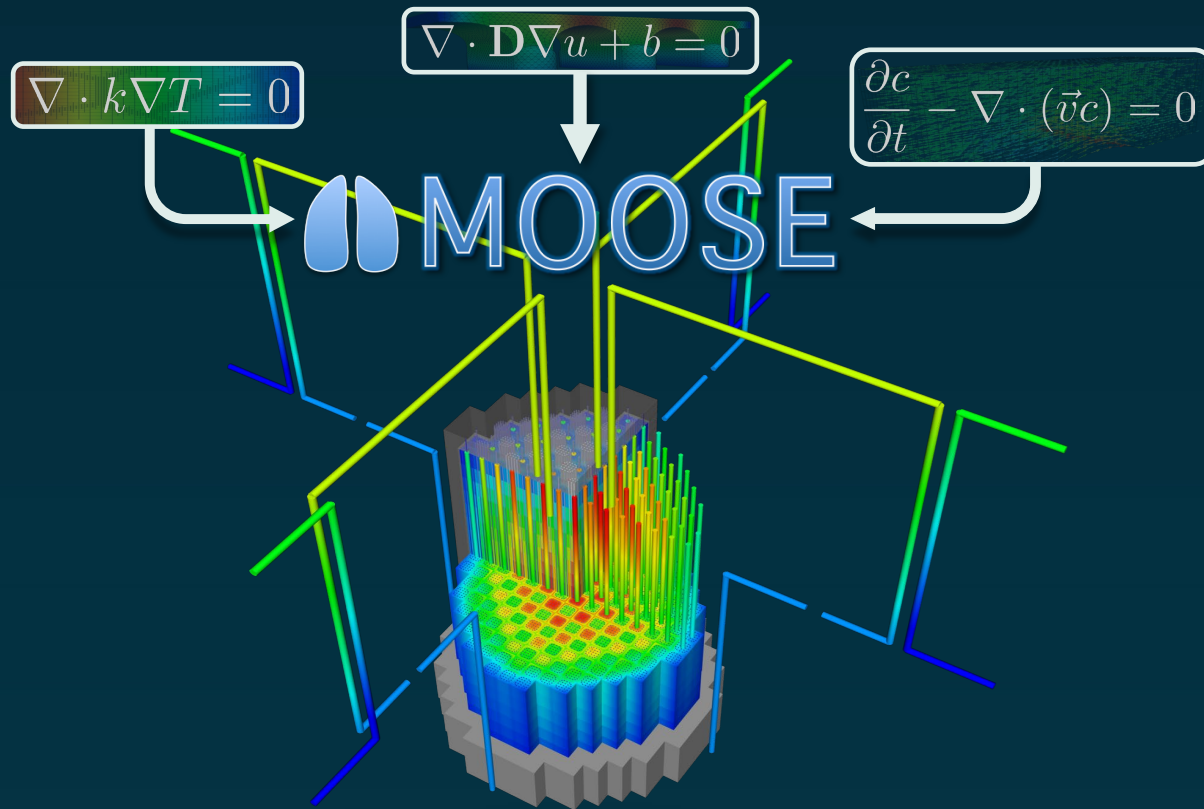
Detailed explanation of models

## 3. Integration with Code Development

Continuous testing of models against codes using CIVET



# MOOSE Framework



## What is MOOSE?

- Multiphysics framework to capture various reactor physical phenomena (neutronics, hydraulics, heat conduction, fuel performance etc.)
- Finite element based (with new finite volume capability) and massively parallelizable
- Standardized input structure
- Nuclear Software Quality Assurance (SQA) compliant

## Why Multiphysics?

- Reactors are inherently Multiphysics
- Upcoming reactor designs rely on non-traditional Multiphysics effects (e.g. core expansion, heat pipes, molten fuel, fluidized beds, porous flow, etc.)
- Predictive Multiphysics enables the study of:
  - Power uprates, material changes, design/safety basis, accident scenarios, design feasibility, moderator choice, fuel loading optimization, material degradation, etc.

# Tutorial

1. Introduction to Multiphysics Coupling with MOOSE
2. Important Terminology and Hierarchy Rules
3. Applications of MultiApps
4. Workflow to Establish a MultiApps Simulation
5. MultiApps Syntax & Examples
6. Transfers Syntax & Examples
7. Advanced Topics
8. Selected FAQ and Other Resources

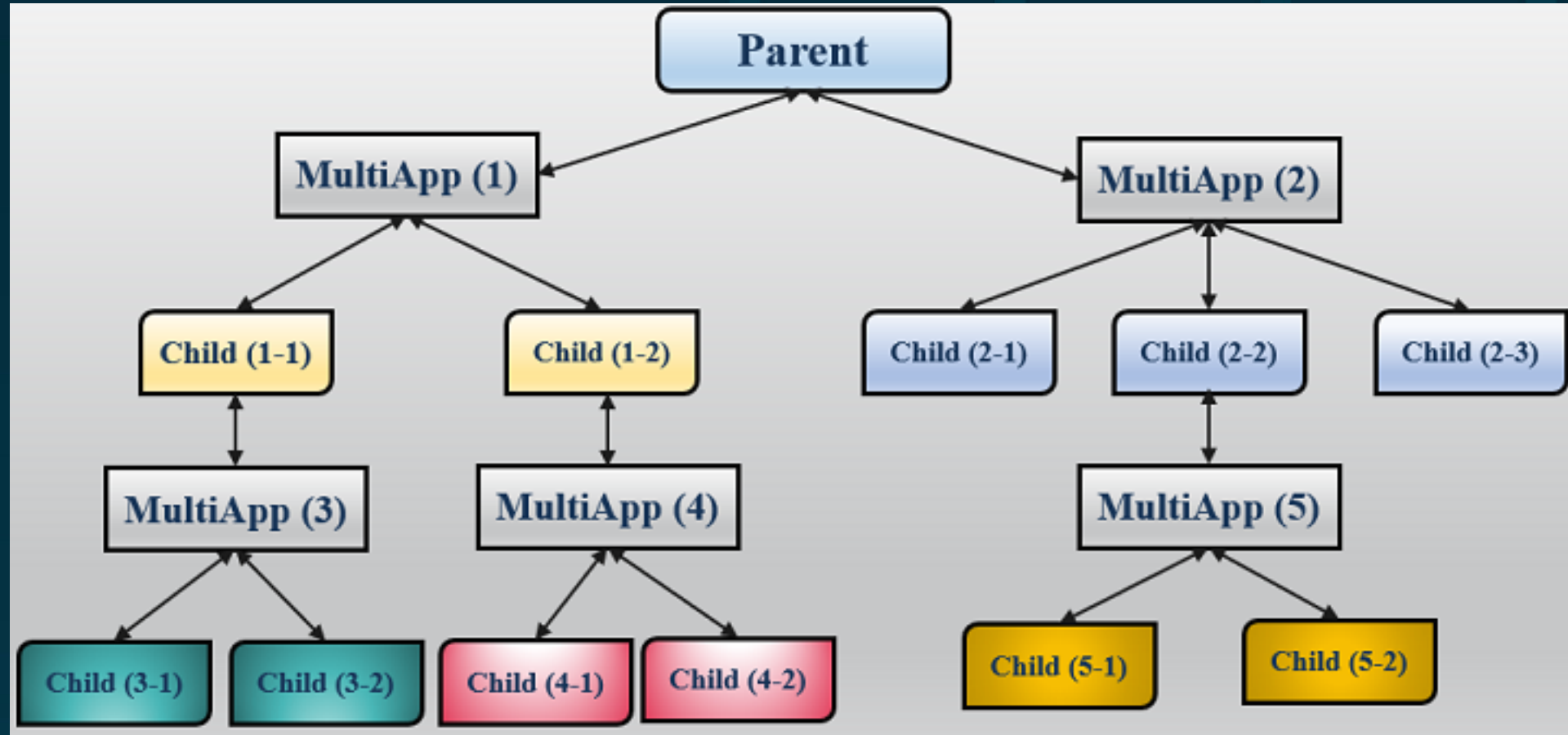


# 1) MultiApps Introduction

- System to numerically separate a physics solve
- Why separate?
  - Dissimilar time scales
  - Dissimilar spatial scales or dimensions
  - Weakly coupled physics
  - Numerical reasons
  - UQ/Optimization studies

## 2) MultiApps Terminology

- Application
- Parent app
- MultiApp
- Child or subapp
- Transfers
- Fixed point



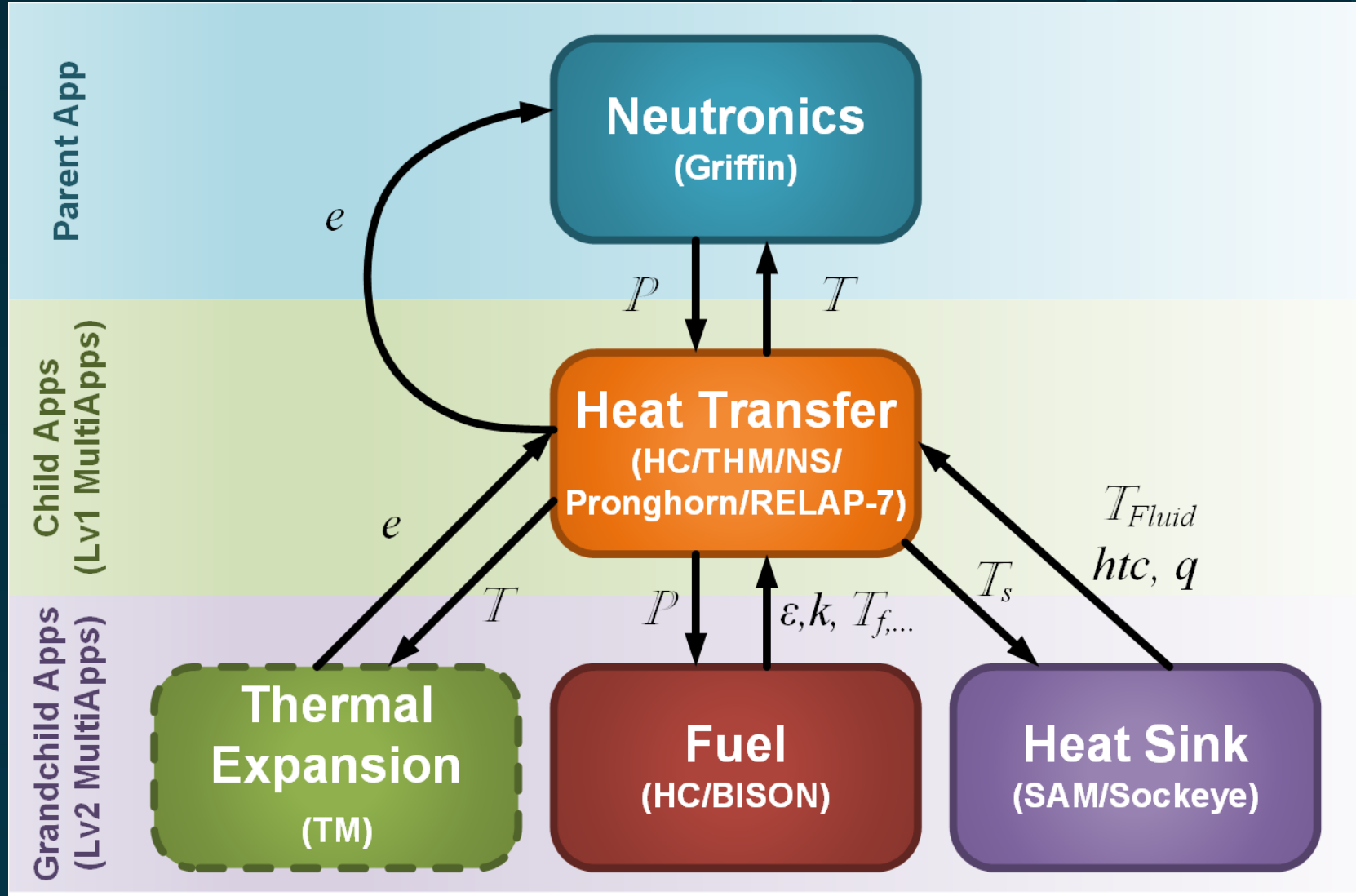
# 3) Applications for MultiApps

- Coupling physics applications from the NEAMS program
- Full list on NEAMS website

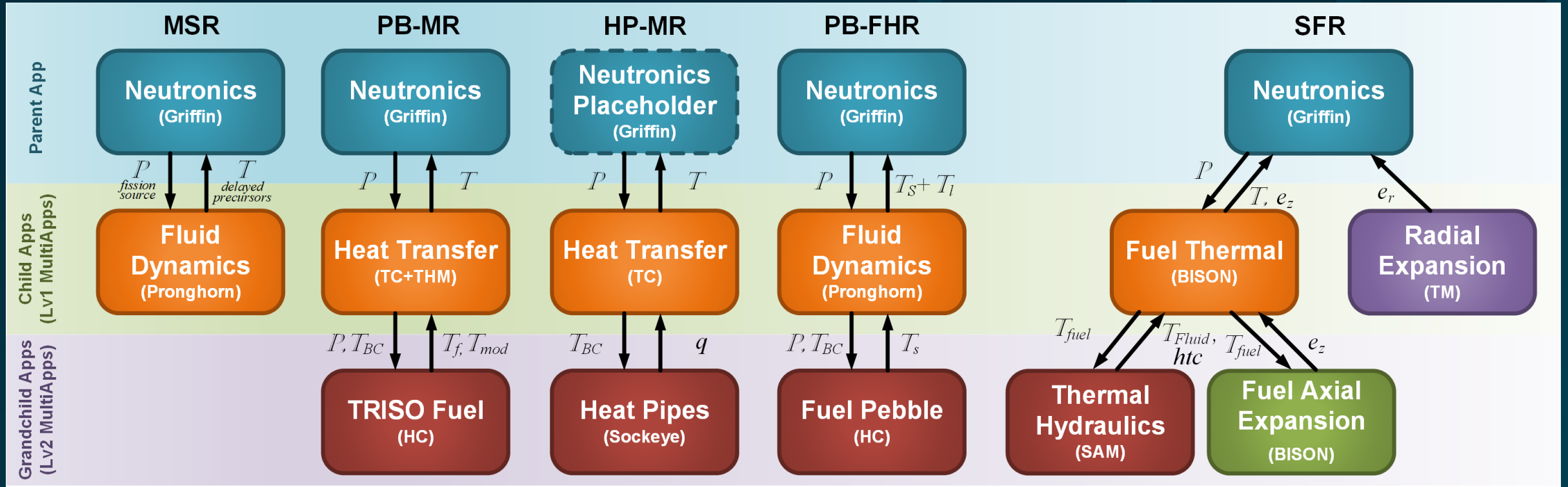
Physics	MOOSE Application	Features
Fuel Performance	<b>BISON</b>	A finite element-based nuclear fuel performance code applicable to a variety of fuel forms including light water reactor fuel rods, TRISO particle fuel, and metallic rod and plate fuel. It is a multiphysics fuel analysis tool that solves fully-coupled thermomechanical problems.
Neutronics	<b>Griffin</b>	A reactor physics application for advanced reactor multiphysics modeling and simulation.
Structure Analysis	<b>Grizzly</b>	A code for modeling degradation of nuclear power plant systems, structures, and components due to exposure to normal operating conditions.
Thermal Hydraulics	<b>nekRS</b>	An open-source Navier Stokes solver based on the spectral element method targeting classical processors and hardware accelerators like GPUs. Use of <b>nekRS</b> in the MOOSE ecosystem is through <b>Cardinal</b> , a wrapping of the spectral element code CFD NekRS and the Monte Carlo radiation transport code OpenMC as a MOOSE application.
Thermal Hydraulics	<b>Pronghorn</b>	A multiscale thermal-hydraulic (T/H) application developed by the Idaho National Laboratory
System analysis	<b>RELAP-7</b>	A next generation nuclear systems safety analysis code being developed at the Idaho National Laboratory (INL).
System Analysis	<b>System Analysis Module (SAM)</b>	A modern system analysis tool being developed at Argonne National Laboratory for advanced non-LWR safety analysis. It aims to provide fast-running, whole-plant transient analyses capability with improved-fidelity for SFR, LFR, and MSR/FHR.
Heat Pipe	<b>Sockeye</b>	A heat pipe analysis application based on the MOOSE framework.
Chemistry	<b>MOSCATO</b>	The <b>MO</b> lten <b>S</b> alt <b>C</b> hemistry <b>A</b> nd <b>T</b> ransp <b>O</b> rt (MOSCATO) code is a code being developed at Argonne that focuses on simulating chemical and electrochemical reactions in molten salts. <b>(MOOSE wrapping development is on-going?)</b>
Chemistry	<b>Mole</b>	An engineering scale MOOSE-based molten salt species transport code developed at Oak Ridge National Laboratory.

### 3) Applications for MultiApps

- Typical core multiphysics model



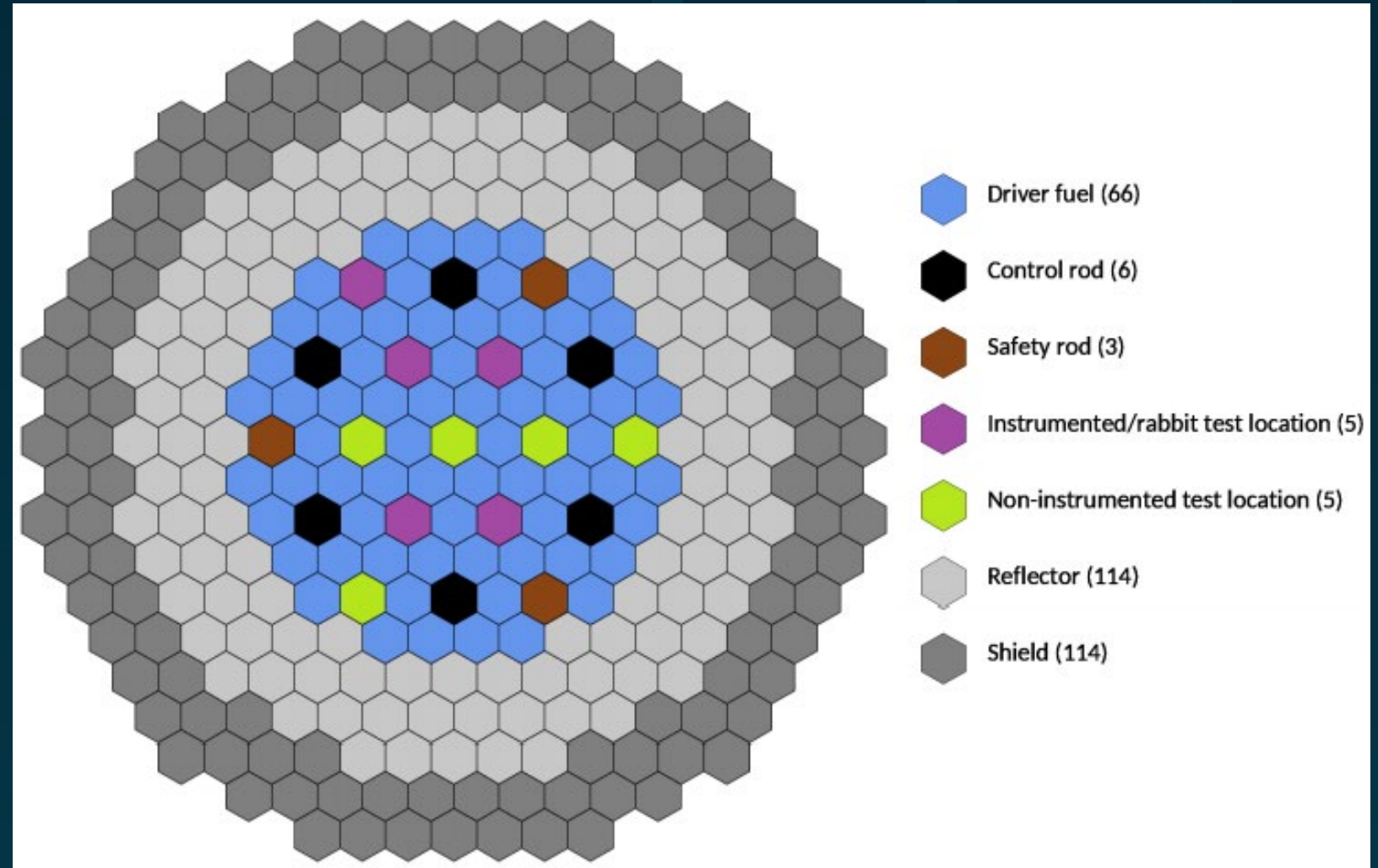
### 3) Multiphysics Models Leveraging MultiApps





## 4) Workflow for Building Multiphysics Application

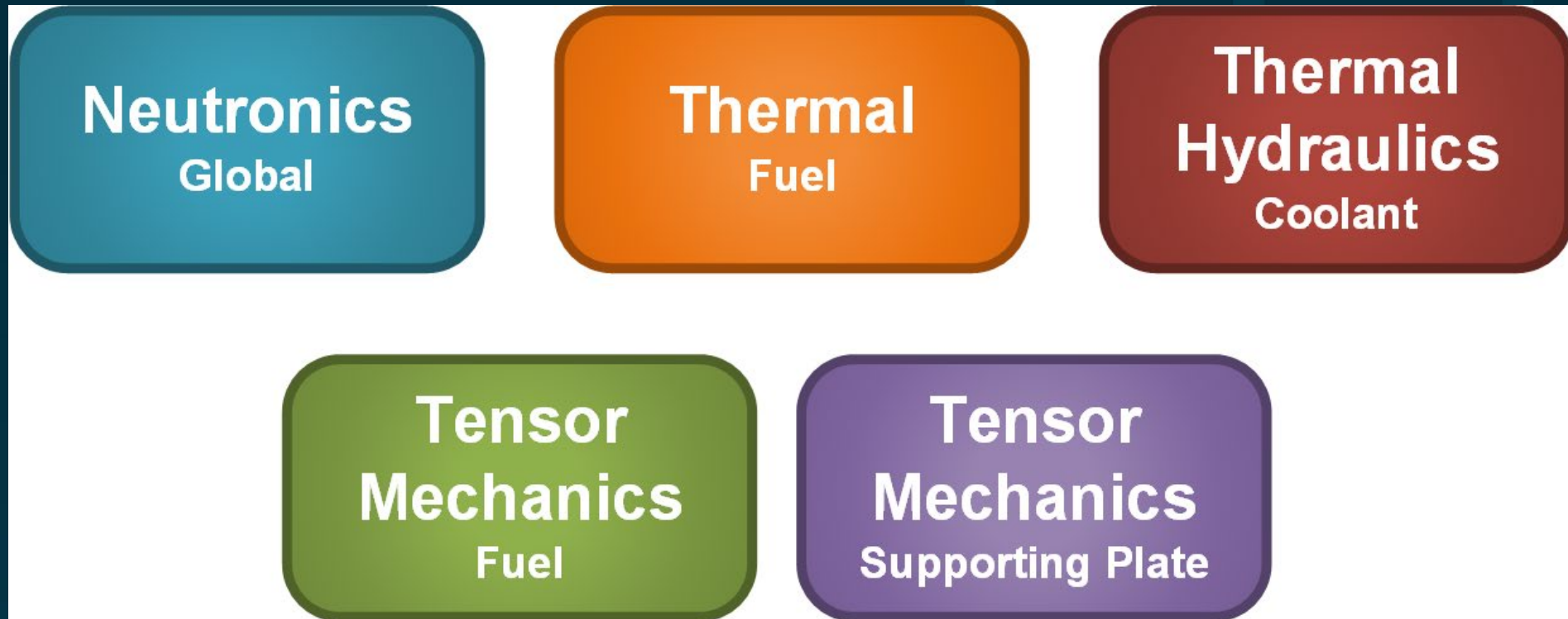
- Sodium Fast Reactor (SFR) example
- Declined in an assembly and a core model, both on VTB
- Contributed to the VTB by N. Martin, T. Folk and VTR team



Radial configuration of the VTR core layout, taken from (Heidet, F. and Roglans-Ribas, J., 2021).

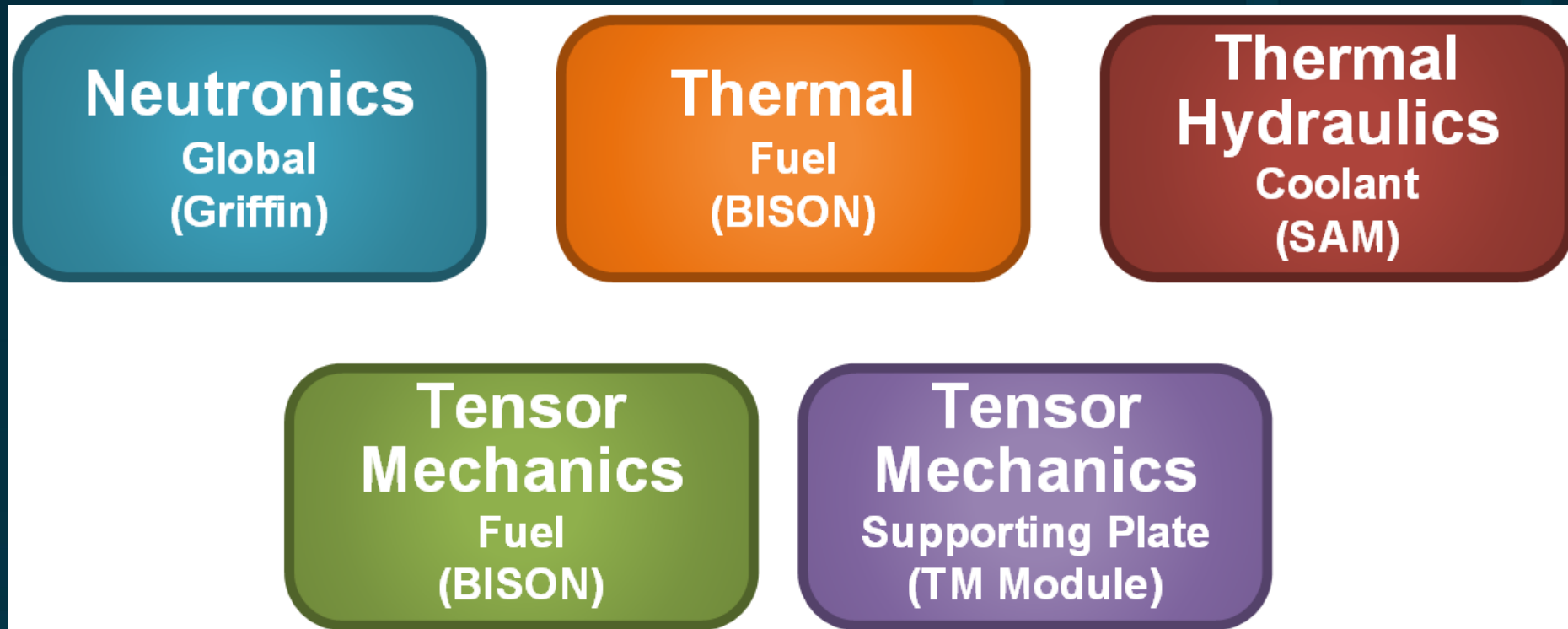
## 4) Workflow for Building Multiphysics Application

- Identify each physics involved



## 4) Workflow for Building Multiphysics Application

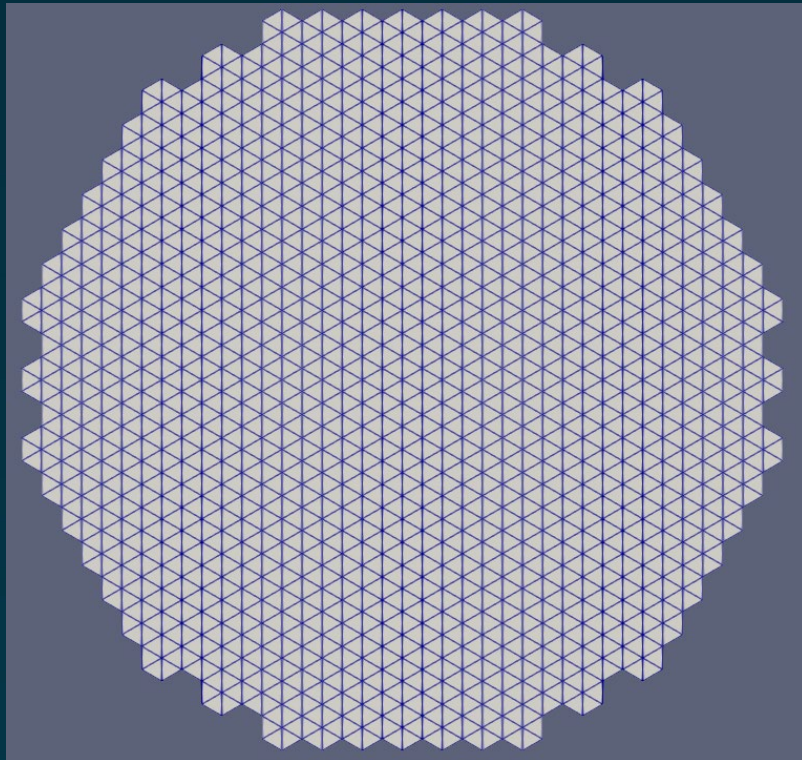
- Identify tool and build relevant single physics model
- Needs to run well standalone



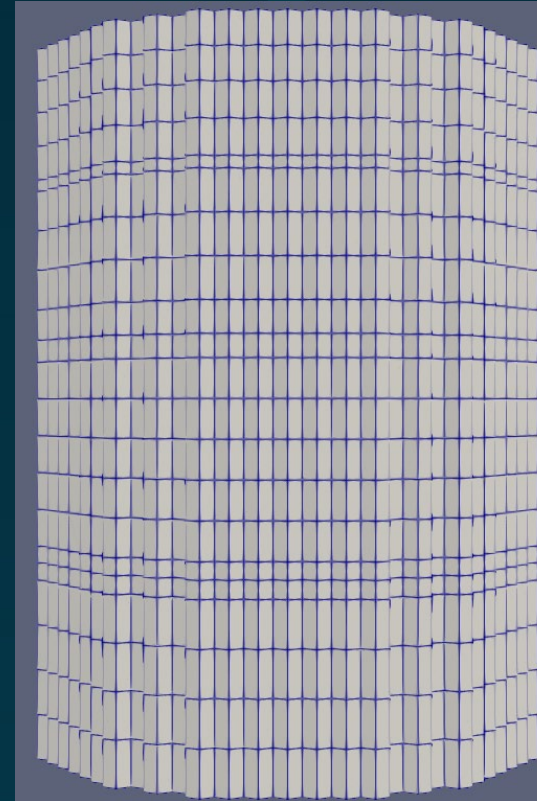


# SFR Example Single-physics Models

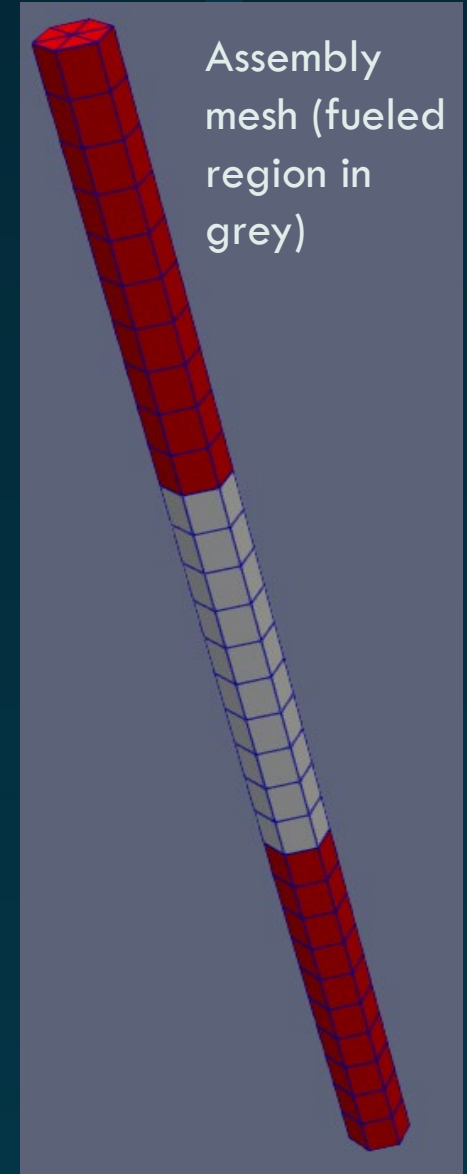
- Single assembly model: homogenized assembly
- Core model: homogenized Continuous Finite Element Method (CFEM) core



Radial mesh  
(left)

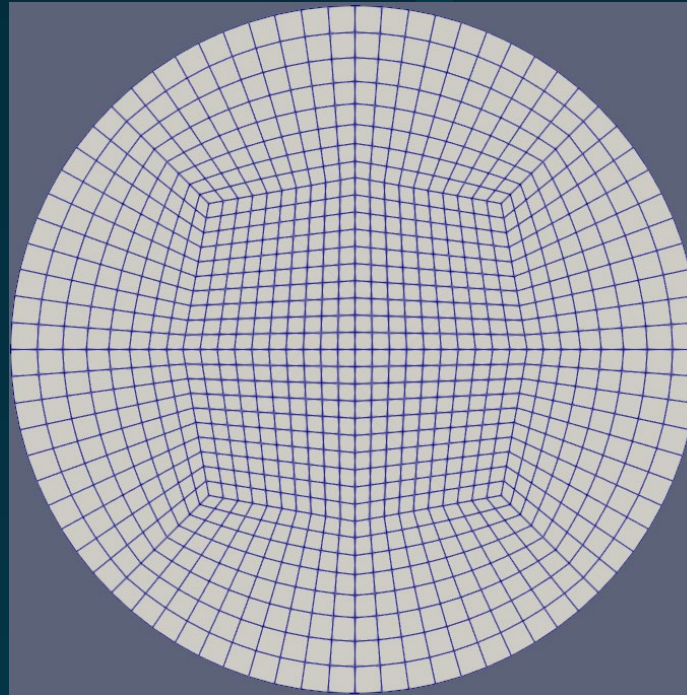


Axial mesh  
(right)

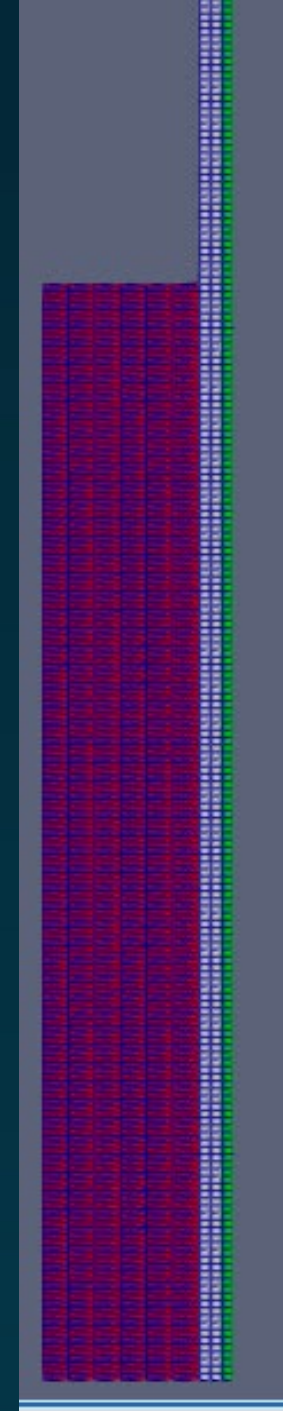


# SFR Example Single-physics Models

- Axial expansion on 2D RZ fuel + clad model
- Radial expansion on 2D support plate



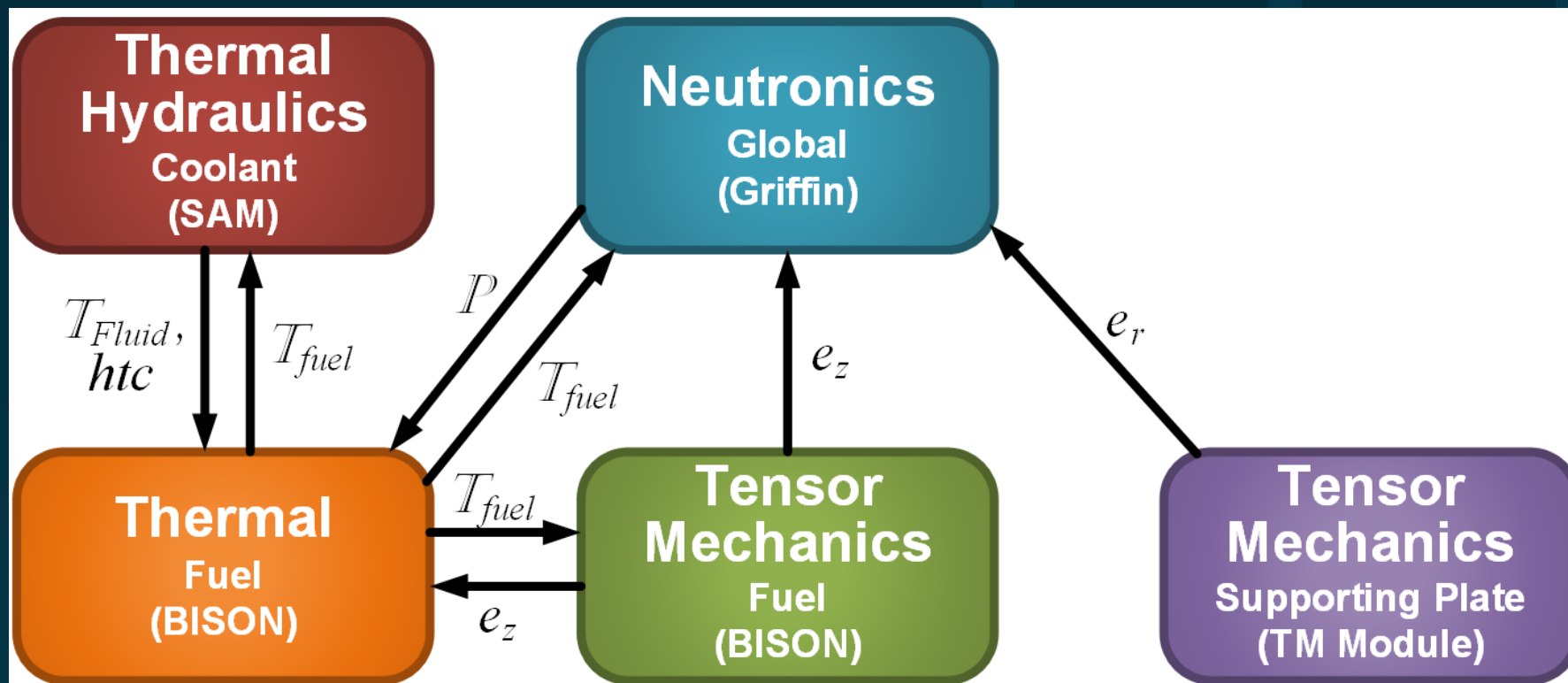
2D support plate



Axial pin  
Model  
red: fuel  
white & green:  
cladding

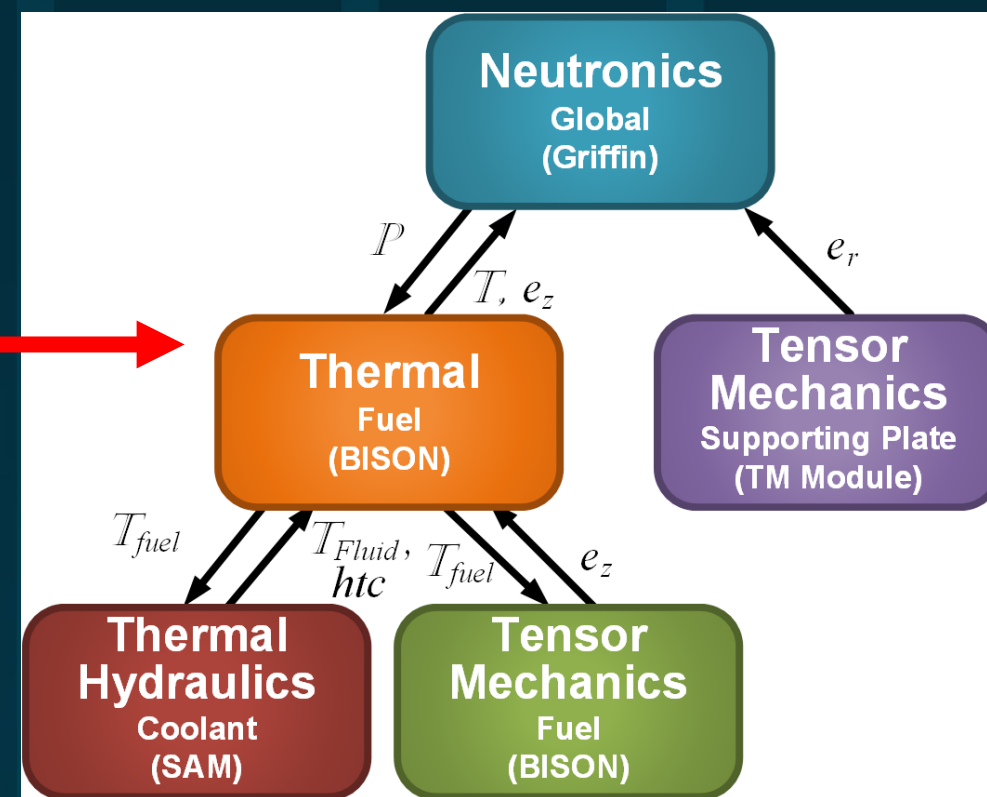
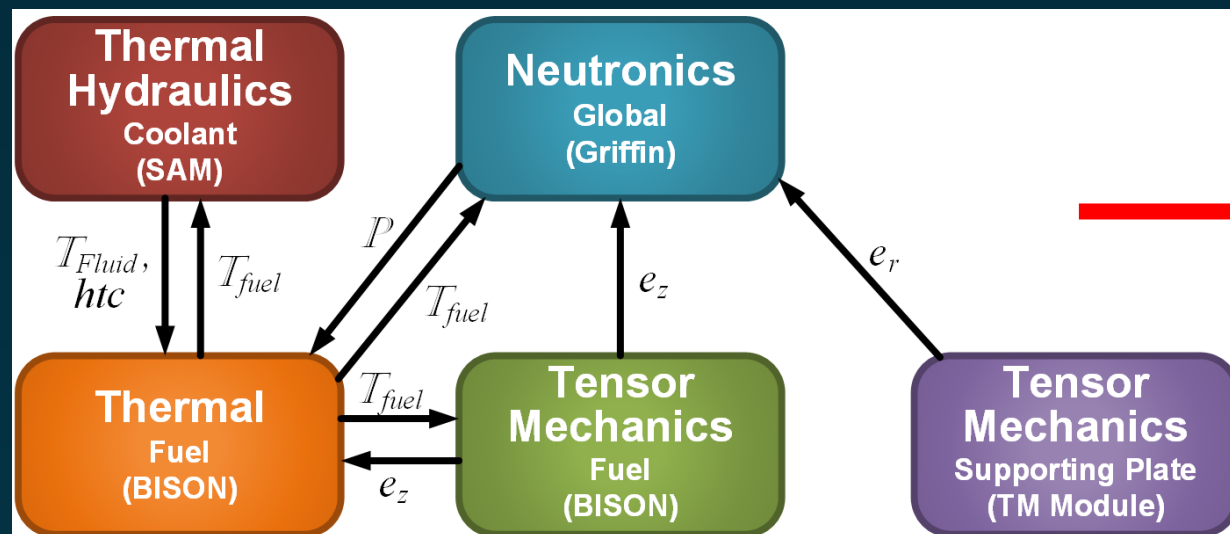
## 4) Workflow for Building Multiphysics Application

- Identify quantities involved in coupling physics



## 4) Workflow for Building Multiphysics Application

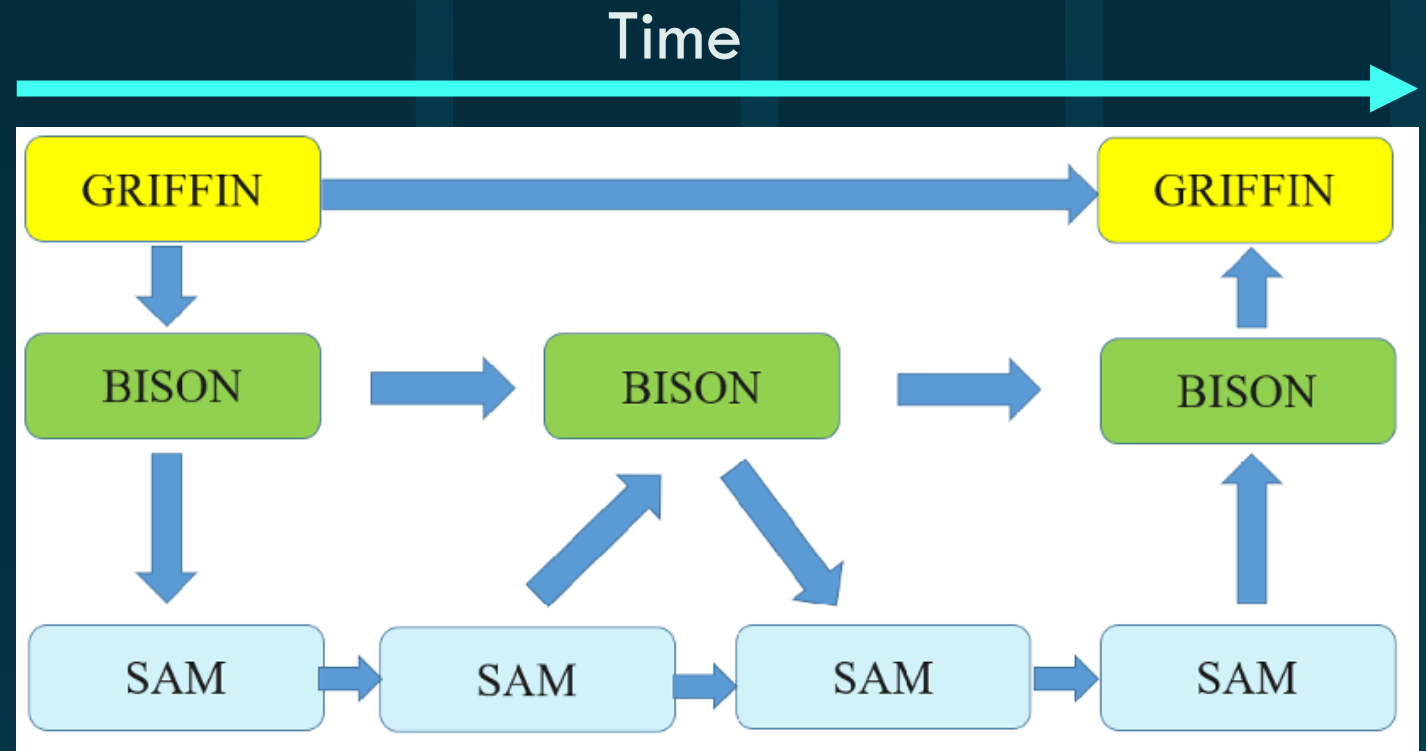
- Determine MultiApps hierarchy





## 5) MultiApps Timestepping

- Default behavior:  
minimum timestep across apps forces synchronization
- Sub-cycles  
Allows subapps to take smaller timesteps  
→ **often main criteria for which app is the main app**
- Catch-up  
Default + some crash-specific behavior

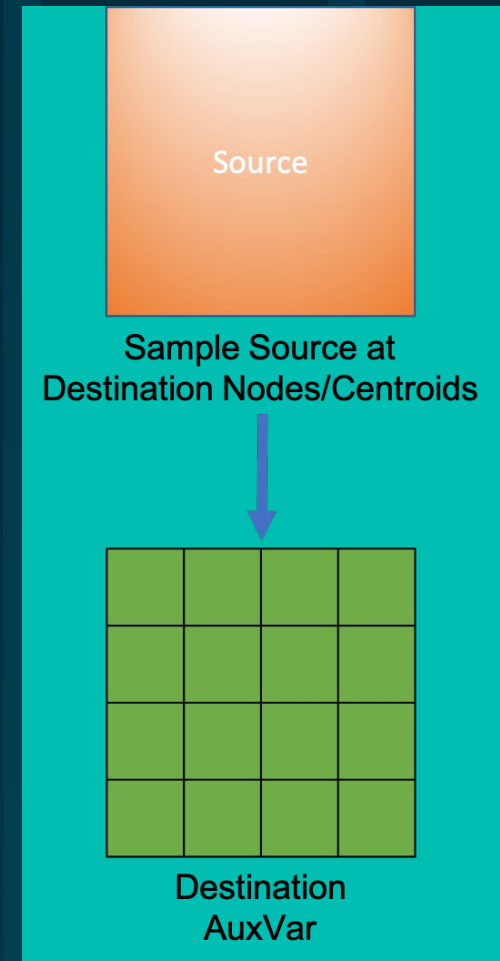


## 5) Other Features & Examples in the Tutorial

- Parallel capabilities and scheduling
- Nesting steady state solves
- Execution order
- Restart and recovery of simulations

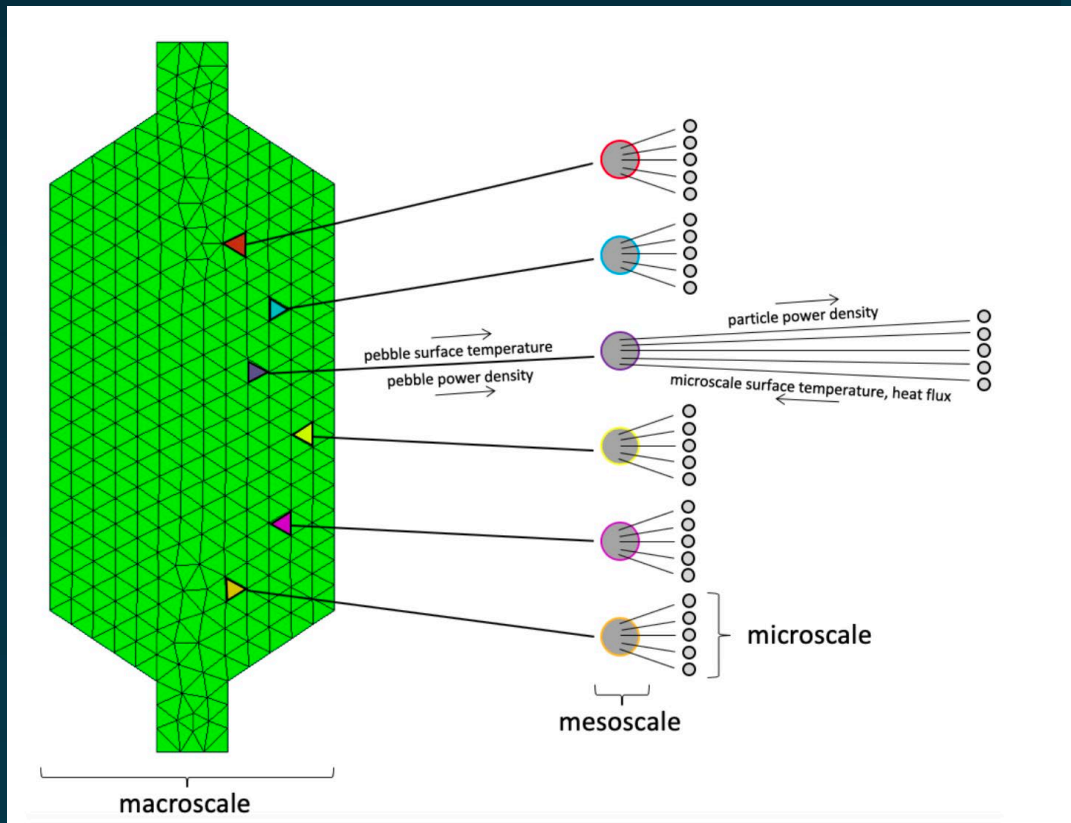
## 6) Transferring Information Between MultiApps

- Fields (spatially dependent)
  - Geometric interpolation
  - Shape function evaluation
  - Projection
  - Straight copy
- Scalar (one or few global numbers)
  - Postprocessors : `MultiAppPostprocessorTransfer`
  - Scalar variables: `MultiAppScalarToAuxScalarTransfer`
  - Reporters: `MultiAppReporterTransfer`

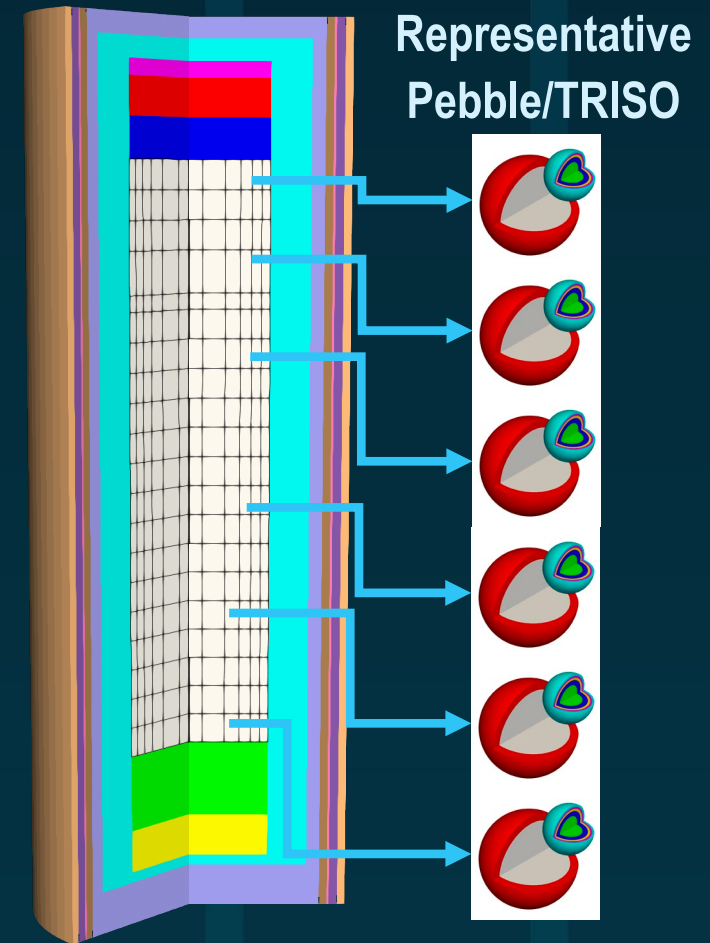


## 6) Transferring Information Between MultiApps

- Conversion between types for multiple child apps and single main app



### Pronghorn model



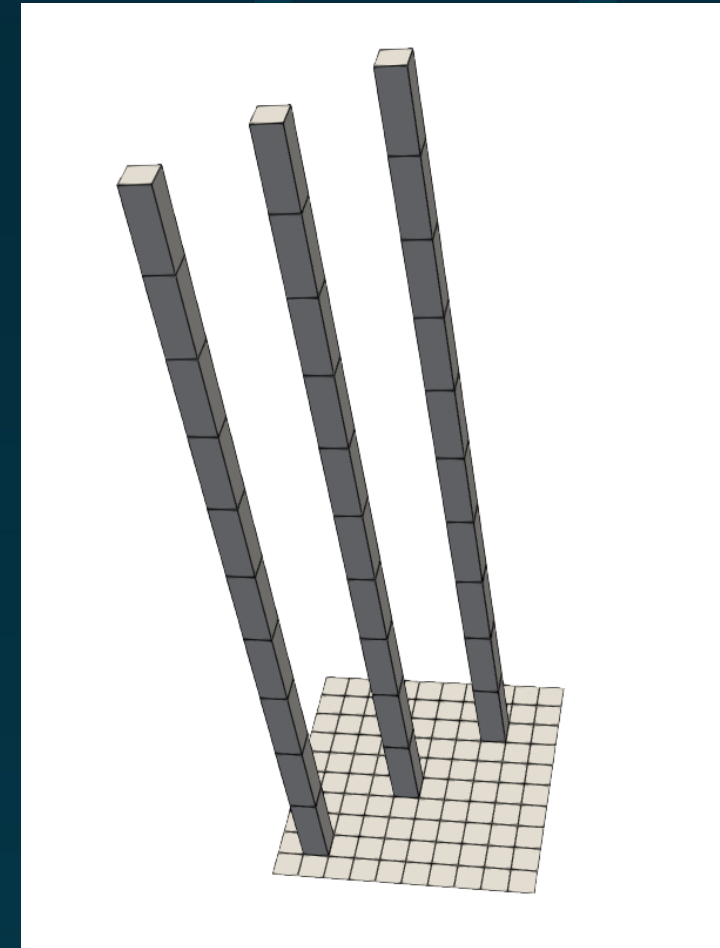
Credit: P. Balestra, Pronghorn (top)  
A. Novak Ph.D. (left)



## 6) Transferring Information Between MultiApps

- SFR example
- Fields:

Displacement: geometric interpolation & copy



## 6) Transferring Information Between MultiApps

- SFR example

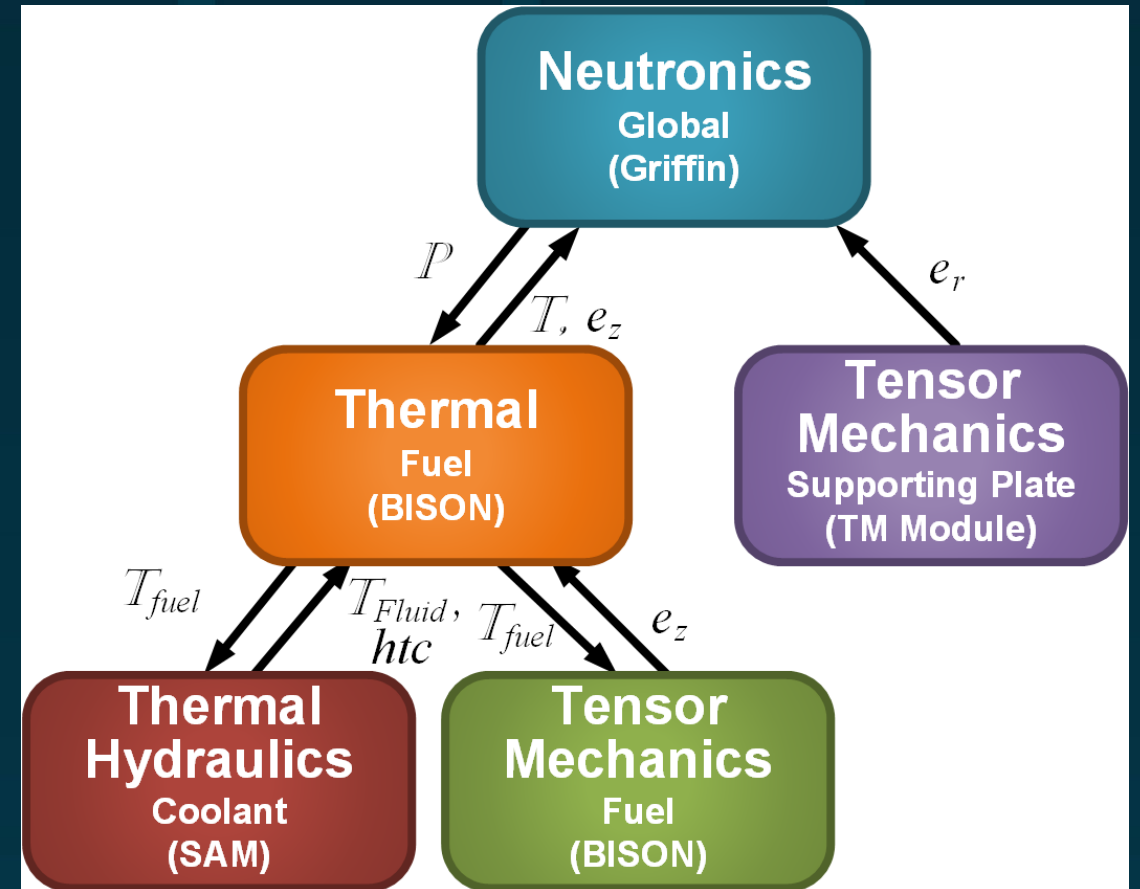
- Fields:

power\_density: projection + conservation

displacement: geometric interpolation & copy

temperature: geometric interpolation  
should it be conservative?

Optimizations: block restriction + fixed mesh



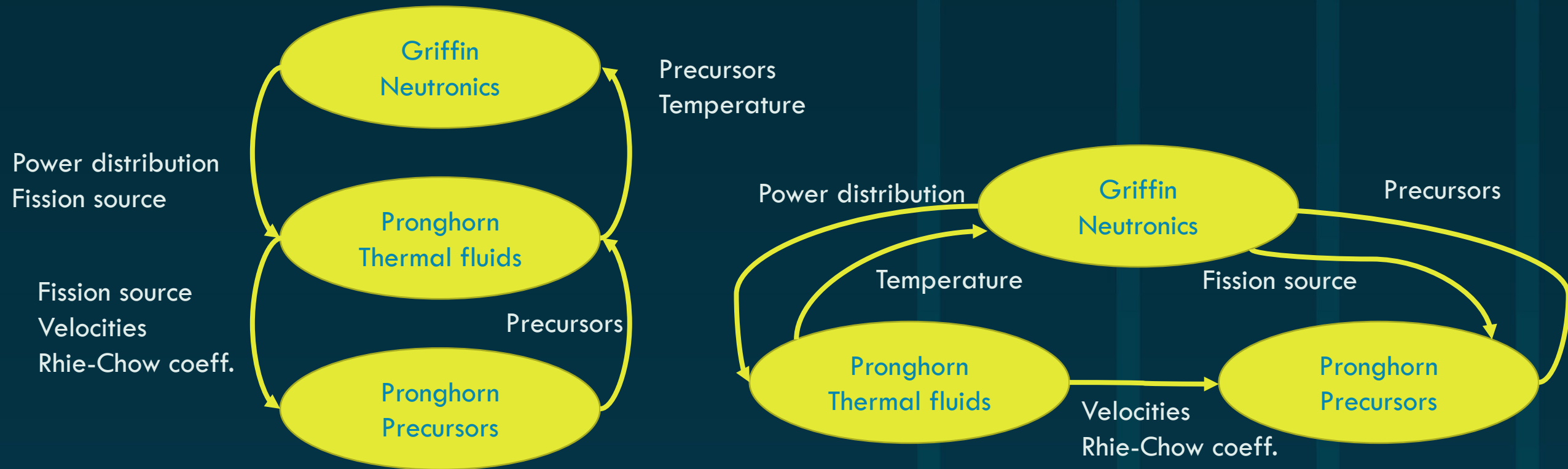
## 6) Transferring Information Between MultiApps

Numerous features:

- Block and boundary restriction
- Fixed mesh mapping caching
- Displaced source/target meshes
- Higher order variables
- Distributed meshes
- Multiple subapps, various parallel configurations
- Conservation of quantities
- Interpolation of transfers during sub-cycling
- Reduction operations (min/max/mean)
- Siblings transfer
- Coordinate transformation

## 6) Transferring Information Between MultiApps

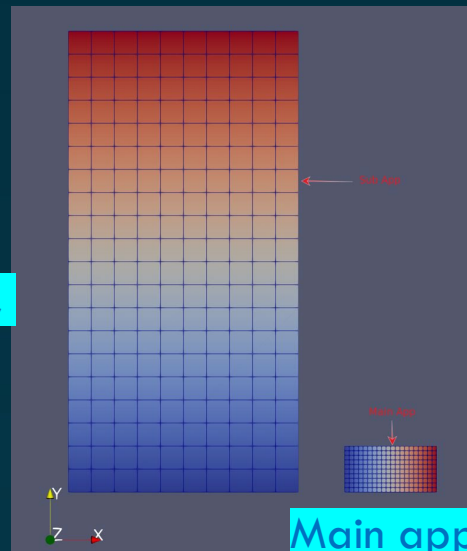
- Siblings transfer implemented in mid-FY22



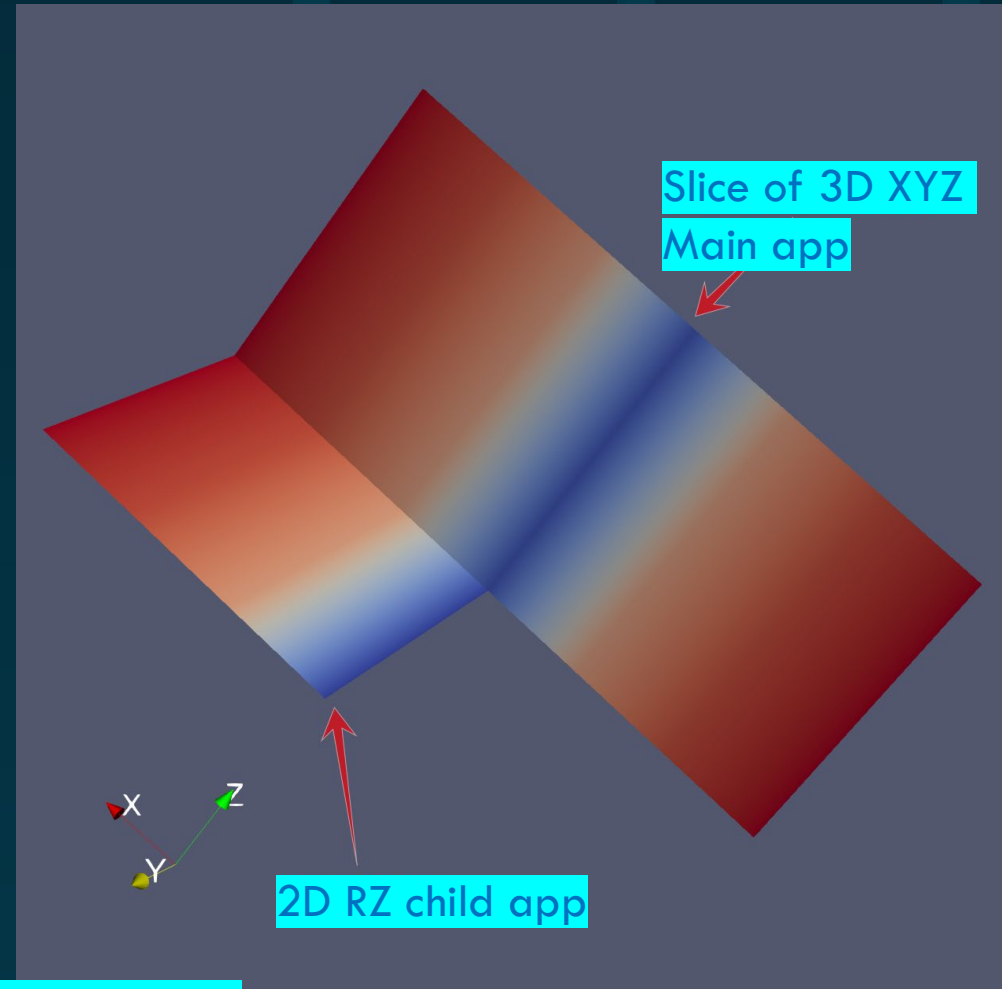
## 6) Transferring Information Between MultiApps

- Coordinate transforms
  - Rotation
  - Scaling
  - Translation (previously supported)
  - Collapsing between coordinates (XYZ, RZ, R-spherical)

Sub app (larger extent,  
scaled using transform)



Main app (small domain)



## 7) Additional Resources (under construction)

- Debugging tips
- Known issues about transfers
  - Floating point comparisons with nearest nodes for example
- Dealing with coupling convergence issues
- Miscellaneous items - will be expanded
  
- Chapter 8 is FAQ

# Summary

- MultiApps are a very powerful and popular system in MOOSE
  - Splits each physics in a different input
  - Automatically handles parallelism, time stepping, and coupling iterations
  - Numerous native ways to transfer data between apps
- Tutorial on MOOSE website on general MultiApps
- Hosts a tutorial on using MultiApps for building nuclear reactor models
  - Uses the SFR (VTR) coupled multiphysics model
  - Necessitates Griffin, BISON & SAM, requestable at [ncrc.inl.gov](https://ncrc.inl.gov)
- Several additional models on the VTB leverage MultiApps
  - Indexed under “Model Features”
- Virtual Test Bed (VTB) repository at: [https://mooseframework.inl.gov/virtual\\_test\\_bed](https://mooseframework.inl.gov/virtual_test_bed)
- Credit to: A. Abdelhameed, A. Novak, A. Lindsay, D. Gaston, N. Martin for figures