

Implementation of a Constructive Solid Geometry Framework in MOOSE for Improved Monte Carlo Integration

ANL: Kalin Kiesling*, Shikhar Kumar, Emily Shemon

INL: Guillaume Giudicelli, Logan Harbour

ORNL: Tarek Ghaddar, Tara Pandya, Rob Lefebvre, Matt Jessee

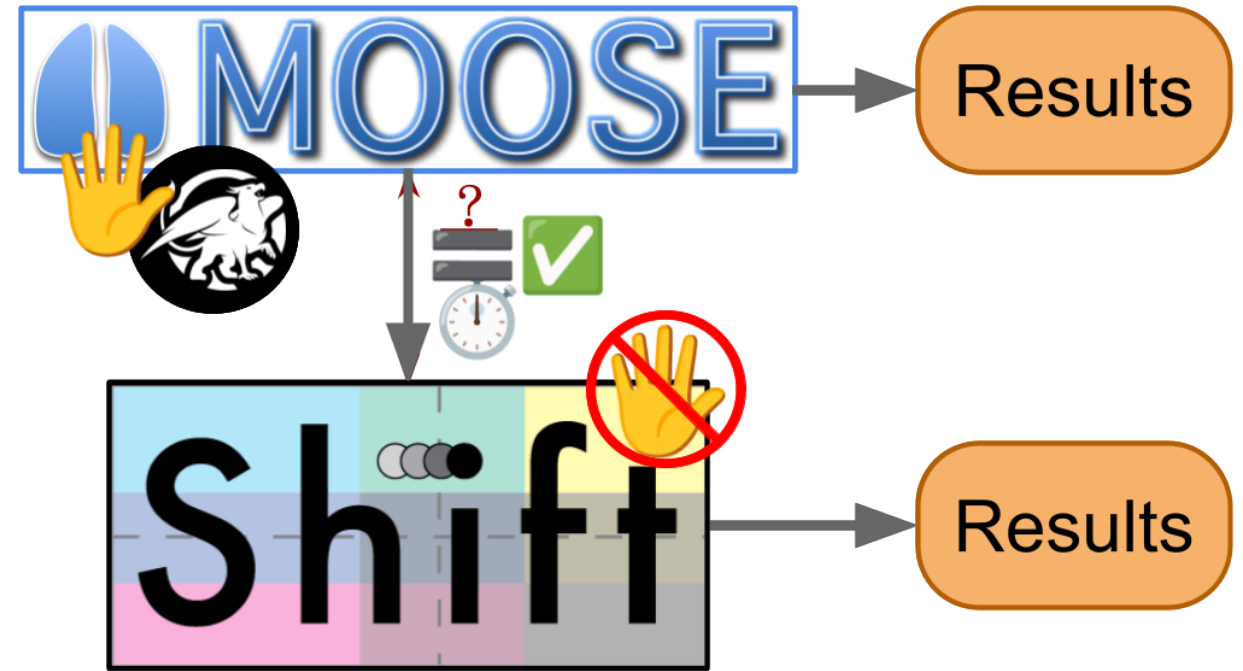


U.S. DEPARTMENT
of **ENERGY**

Office of
Nuclear Energy

Objective

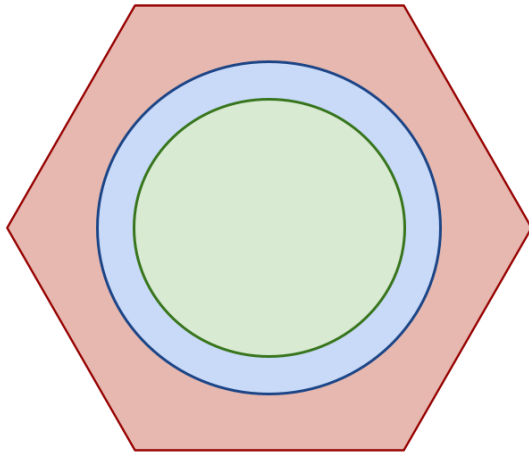
The Constructive Solid Geometry (CSG) framework in MOOSE aims to **automate the geometry connection** between MOOSE and MOOSE-based apps and Monte Carlo (MC) codes for **more reliable and efficient cross section generation, code-to-code comparisons, generation of reference solutions, and multiphysics workflows.**



Constructive Solid Geometry vs. Finite Element Mesh

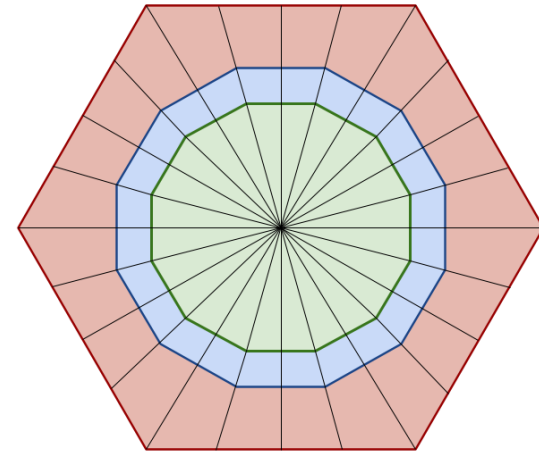
Constructive Solid Geometry (CSG)	Finite Element Mesh (FEM)
• Spatial regions are defined by <u>Boolean combinations</u> of surfaces, regions, cells, and universes. • Surfaces are defined analytically: ➤ <u>Exact</u> representation without approximations. • Most common Monte Carlo geometry representation.	• Spatial regions are defined by <u>discretized</u> surfaces. • Surfaces are a collection of discretized mesh elements: ➤ Some <u>approximation</u> is required, accuracy depends on discretization parameters. • Most common MOOSE geometry representation.

CSG

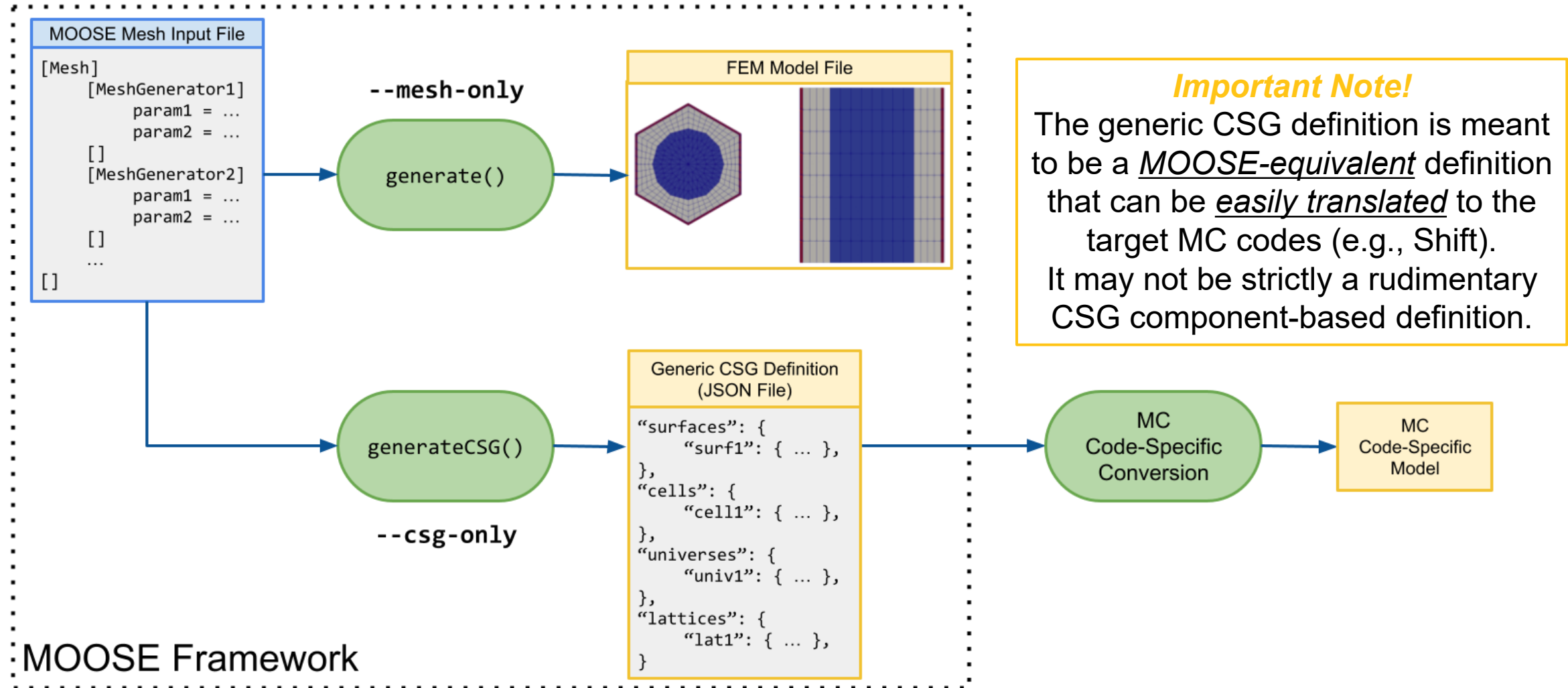


An example of a 2D pin cell using a CSG representation (left) and an exaggerated coarse FEM representation (right).

FEM



MOOSE-to-Monte Carlo User Workflow



Developmental Status

Key

Completed

In-Progress

Future Work

Not Planned

CSGBase Framework

Components

Surfaces

Plane

Sphere

Cylinders

Custom types

Regions

Half-space

Union

Intersection

Complement

Cells

Void

Universe fill

Lattice fill

"Material" fill

Universes

Collections of cells

Lattices

Hexagonal

Cartesian

Custom types

Trans-formations

Rotation

Translation

Scale

Engineering Units

Polygons

Custom types

Mesh Generator (MG)

generateCSG()

Implementation

Reactor Module

Reactor Geometry Mesh Builder (RGMB)

Pin MG

Assembly MG

Core MG

Pin cell Unit

Other Reactor MGs

Framework

Most Common MGs

Code-Specific

Conversion Methods

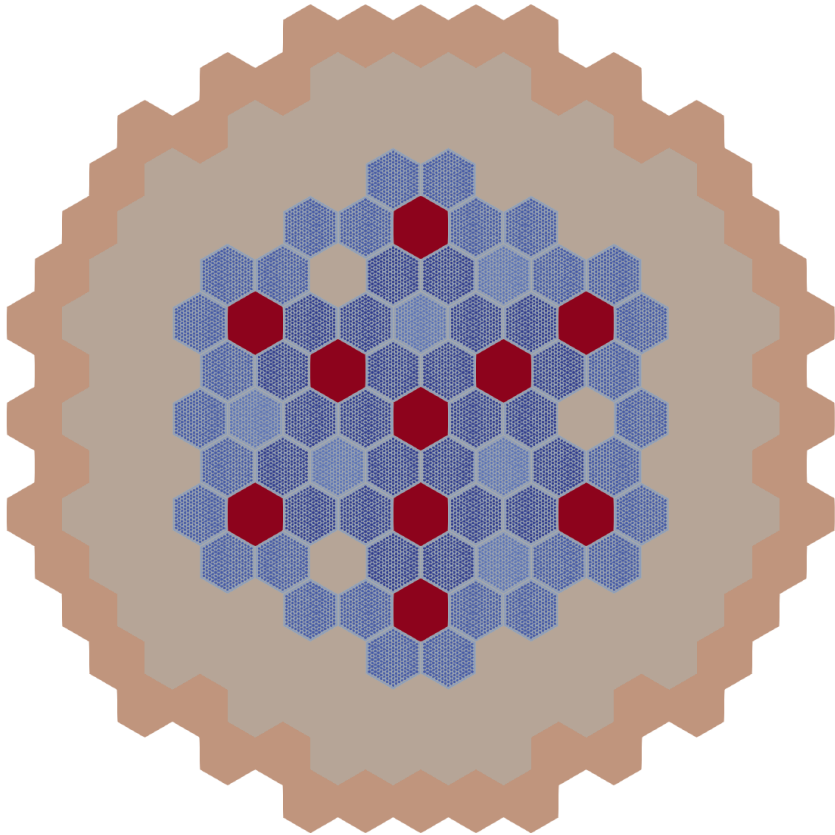
JSON → MC

Titan (Shift)

Other MC Codes

MOOSE MG input (select snippets)

```
[fuel_ass
type
assem
backg
backg
duct_
duct_
duct_
```



```

5 5 4 4 4 4 4 4 4 5 5;
6 5 5 4 4 4 4 5 5 6;
6 6 5 5 5 5 5 6 6'
extrude = true
[]

```

```

${mid_low
${mid_soc
${mid_upp
${mid_upp

```

```
as_plen}'
```

bly

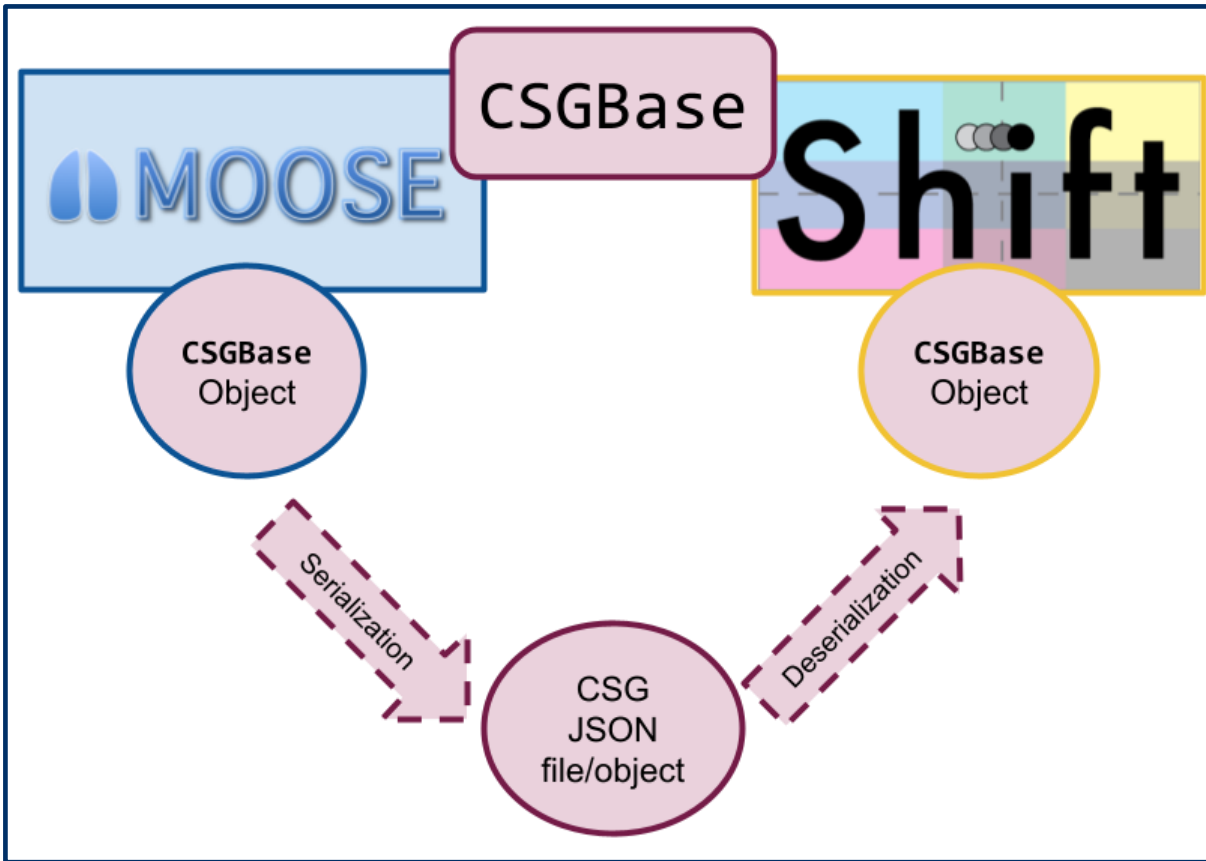
The image displays a complex, multi-colored hexagonal lattice structure, resembling a snowflake or a stylized flower. The structure is composed of many small hexagons arranged in a symmetrical, fractal-like pattern. The colors used include various shades of blue, green, yellow, and red, creating a vibrant, crystalline appearance. The overall shape is roughly circular with a jagged, crystalline edge. The structure is centered on a white background.

```
0",
1",
2",
3",
ell"
```

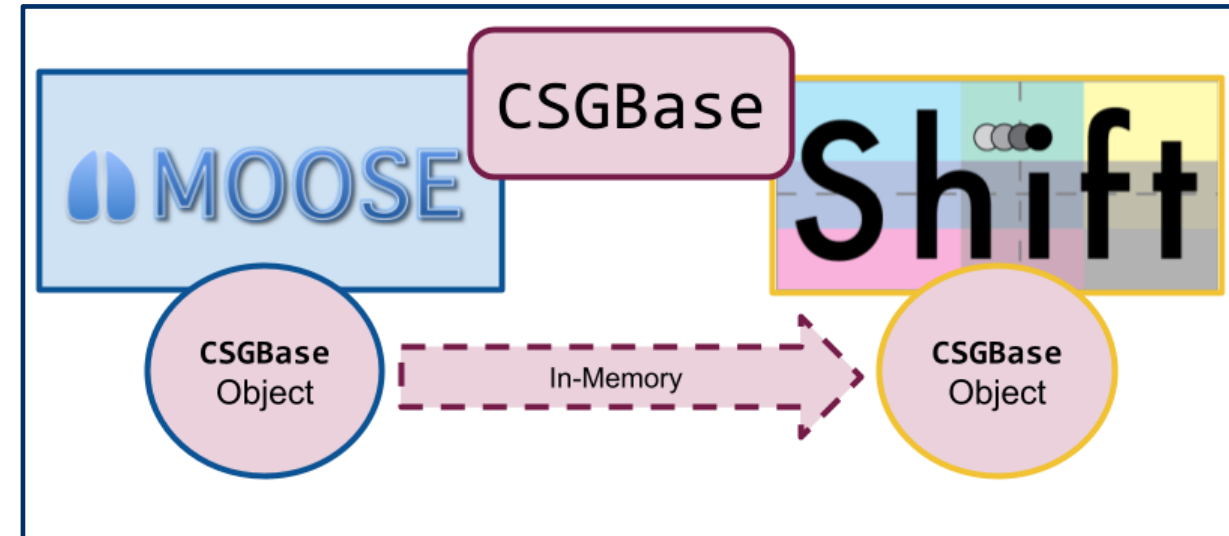
JSON CSG Output (select snippets)

MOOSE-to-MC Code Coupling Options

Option 1: using a file/object layer



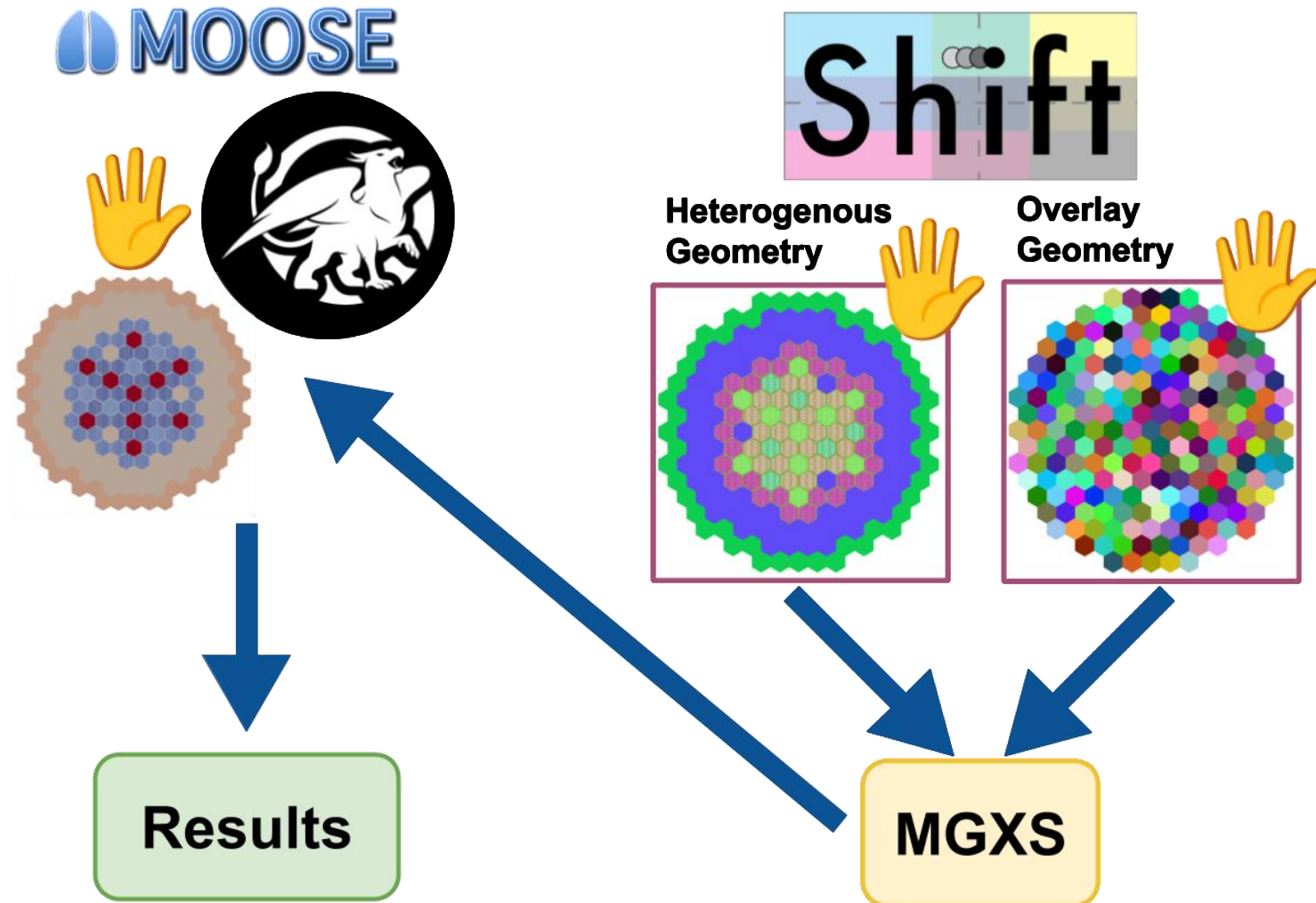
Option 2: direct in-memory coupling



CSGBase will exist as an external module for easier shared functionality.

Example Use Case: Cross Section Generation Using Shift Overlays for Griffin

Current
Manual
Process

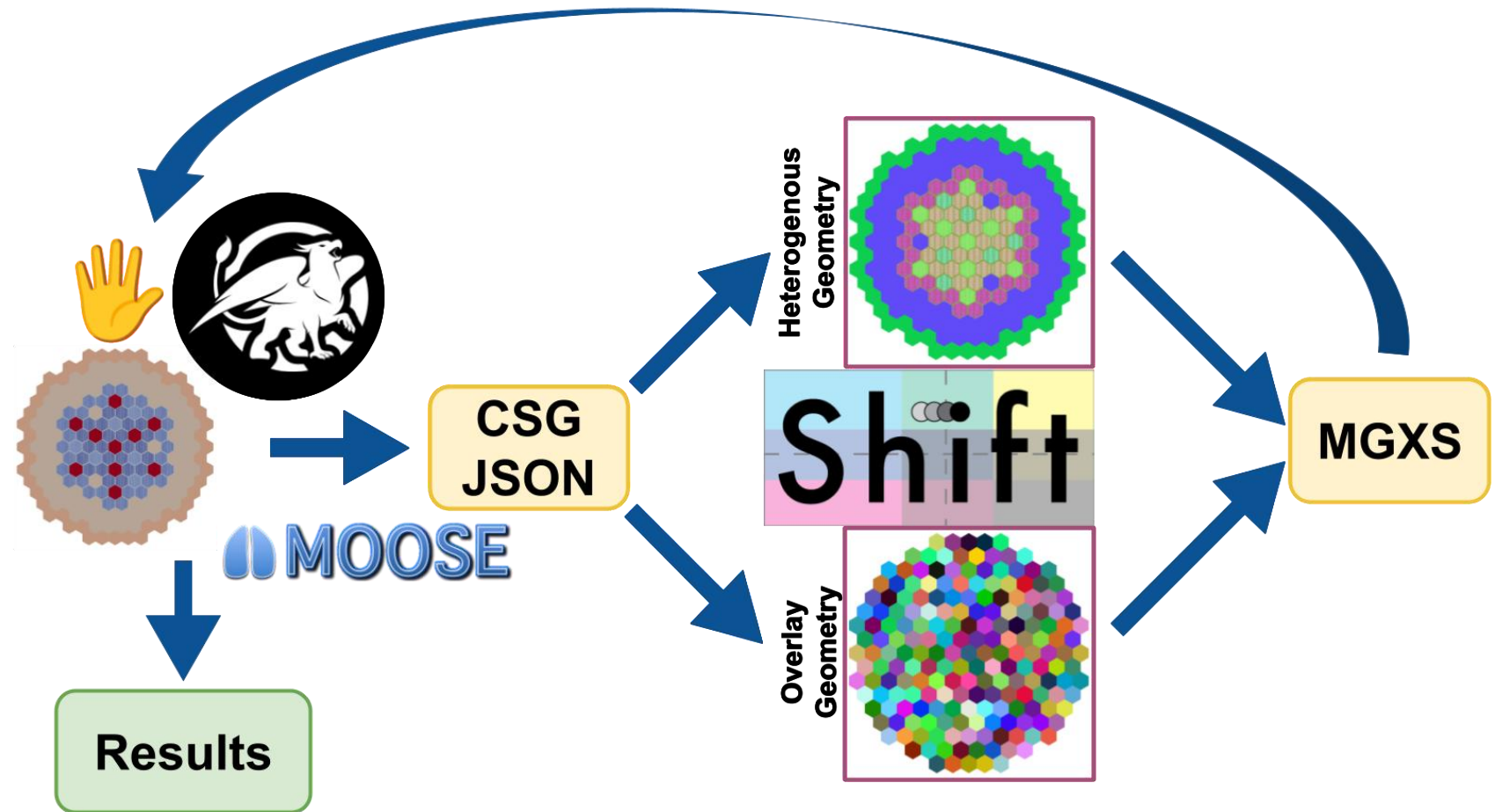


U.S. DEPARTMENT
of **ENERGY**

Office of
Nuclear Energy

Example Use Case: Cross Section Generation Using Shift Overlays for Griffin

Automated
Process
using
MOOSE
CSG



Future Work

- Remainder of FY26:
 - Finish implementation of the **CSGBase framework components** (transformations and engineering units).
 - Implement the **generateCSG() method** for key mesh generators.
 - Implement the initial **MOOSE-to-Shift conversion method** via Titan.
- Additional:
 - Demonstrate MOOSE-to-Shift workflow for a regular fast reactor core geometry.
 - Determine how to support MOOSE-based material definitions and other spatial domain characteristics (e.g., tally regions, depletion zones, overlay geometries, etc.).
 - Advise MC code developers on implementation methods.





U.S. DEPARTMENT
of **ENERGY**

Office of
Nuclear Energy