Joint ICTP-IAEA Workshop on Monte Carlo Radiation Transport and Associated Data Needs for Medical Applications
28 October – 8 November 2024
ICTP, Trieste, Italy

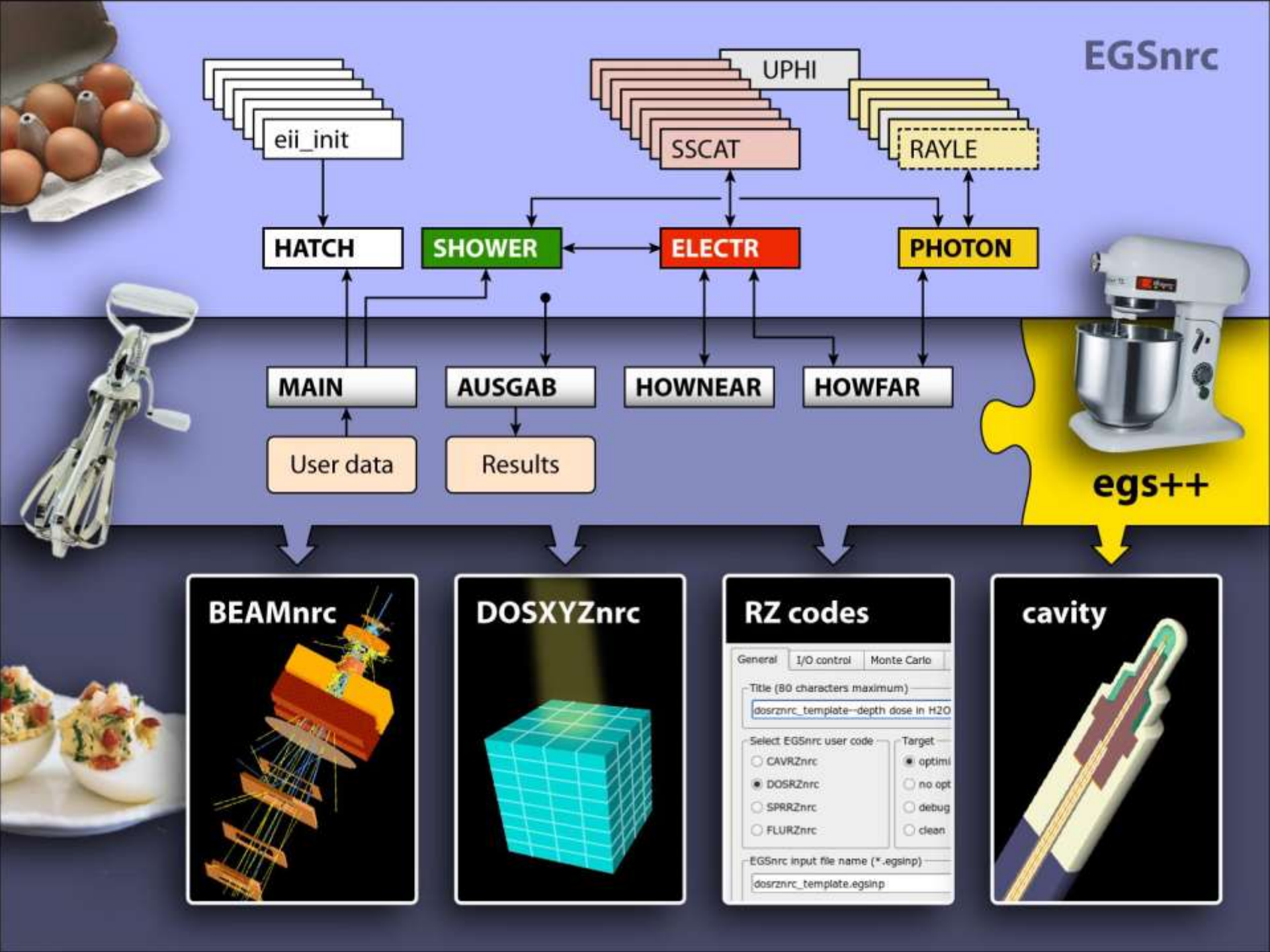
Lecture 4

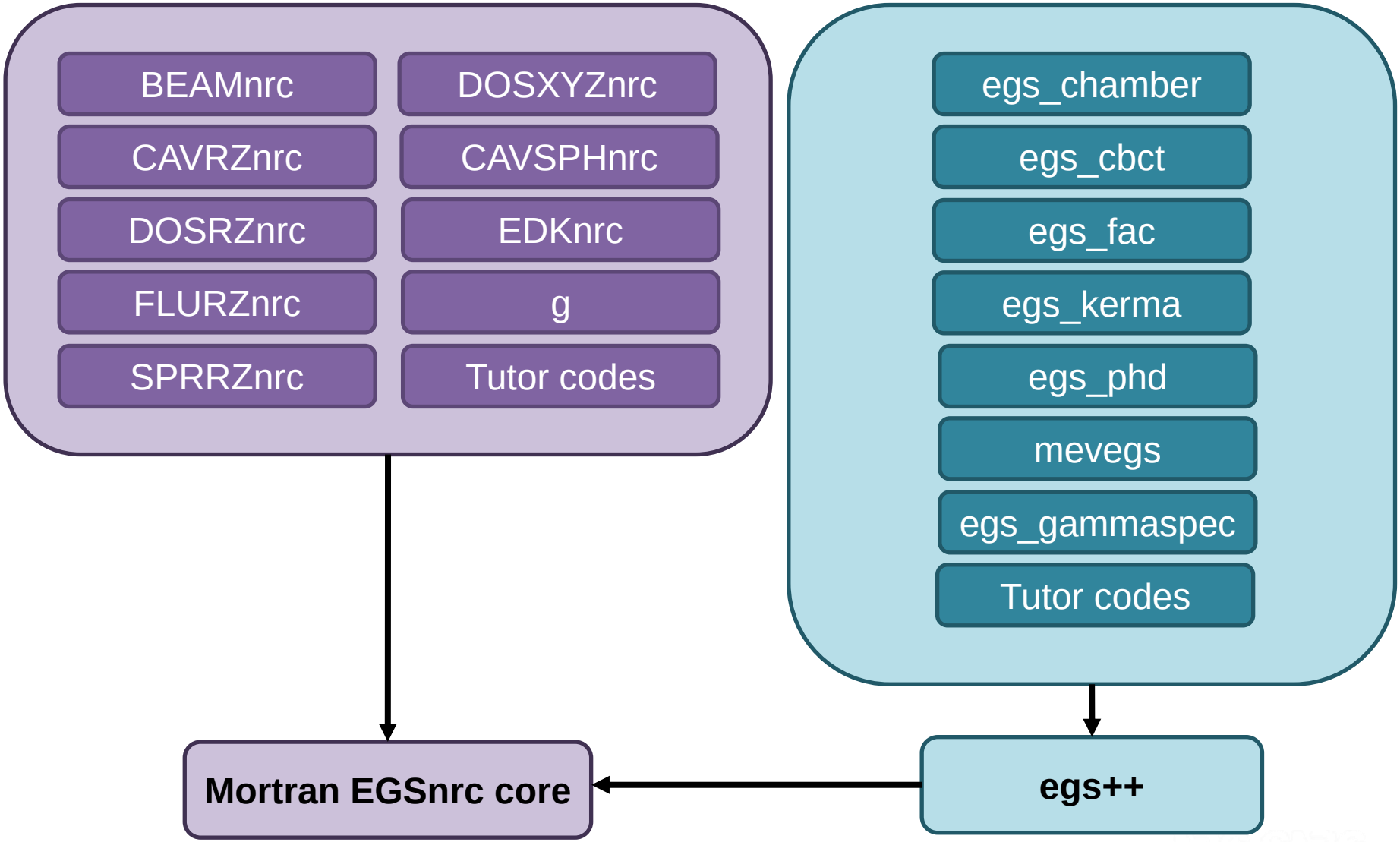
Looking inside EGSnrc

Reid Townson

Metrology Research Centre
National Research Council Canada

Government of Canada | Gouvernement du Canada
Carleton University
Université de Montréal
nova scotia health


EGSnrc
eii_init
UPHI
SSCAT
RAYLE
HATCH
SHOWER
ELECTR
PHOTON
MAIN
AUSGAB
HOWNEAR
HOWFAR
User data
Results
egs++
BEAMnrc
DOSXYZnrc
RZ codes
General
I/O control
Monte Carlo
Title (80 characters maximum)
dosrznrc_template--depth dose in H2O
Select EGSnrc user code
CAVRZnrc
DOSRZnrc
SPRRZnrc
FLURZnrc
Target
optimi
no opt
debug
clean
EGSnrc input file name (*.egsinp)
dosrznrc_template.egsinp
cavity


BEAMnrc
DOSXYZnrc
CAVRZnrc
CAVSPHnrc
DOSRZnrc
EDKnrc
FLURZnrc
g
SPRRZnrc
Tutor codes
egs_chamber
egs_cbct
egs_fac
egs_kerma
egs_phd
mevegs
egs_gammaspec
Tutor codes
Mortran EGSnrc core
egs++

MORTRAN

Looking inside MORTRAN applications

- MORTRAN is simply a **string processor** that produces Fortran77 code
- Primarily this is for code re-use via **macros**

- Application code gets packaged with the EGSnrc core:
  - **egsnrc.macros + yourApp.mortran + egsnrc.mortran**

- The package is processed by **mortran3.f**, converting to a final Fortran file **yourApp.f**

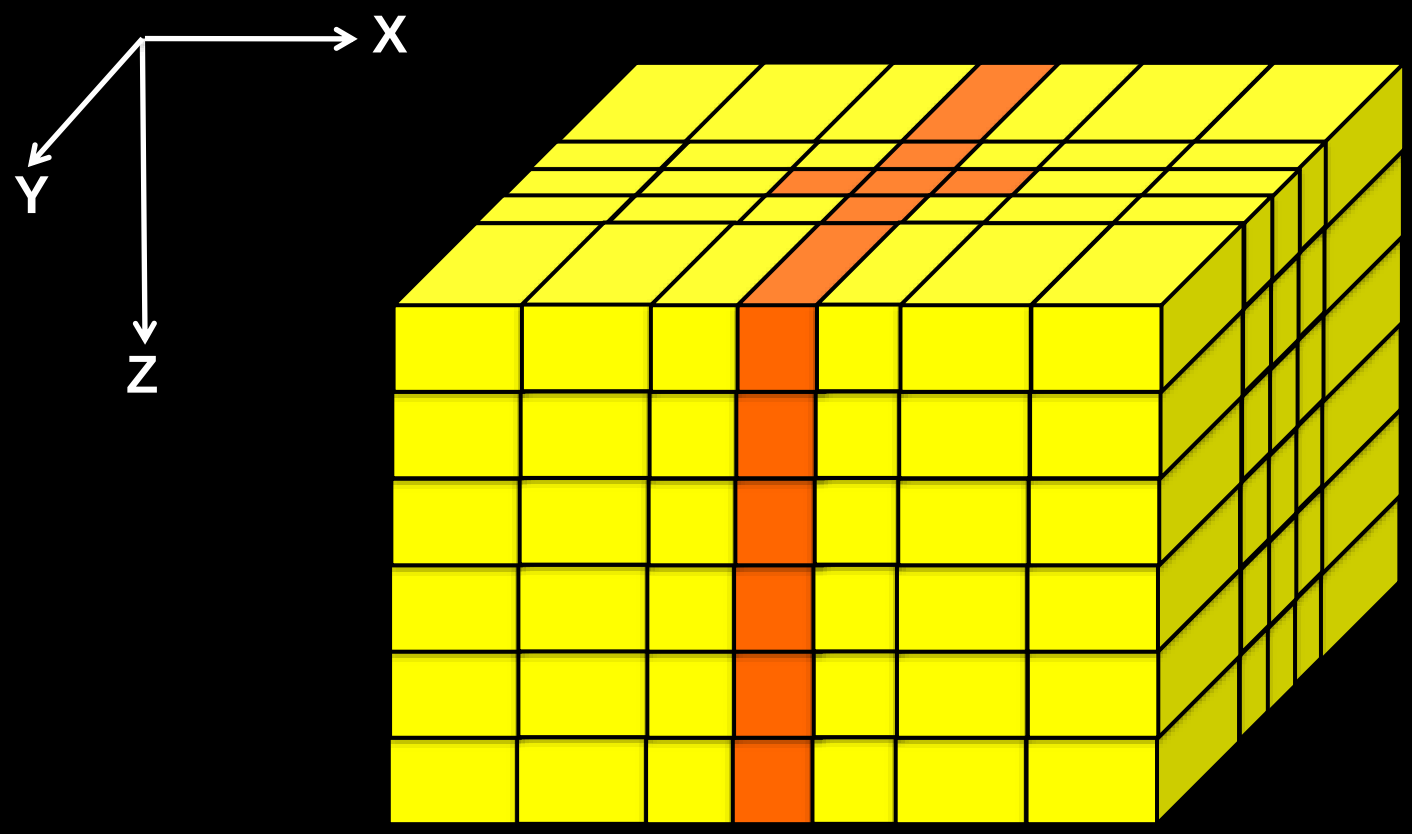
EGSnrc MORTRAN applications don't share*

- Each application must define its own:
  - **Particle sources**
  - **Geometries**
  - **Input/output**
  - **Interaction with EGSnrc core**
    - **initialization, shower() calls, etc.**

*except when they do

DOSXYZnrc: dose in voxels

- 22 sources
- 1 geometry type
- Unique input/output



DOSXYZnrc gets packaged

```
SOURCES = $(EGS_SOURCEDIR)egsnrc.macros \
          $(EGS_UTILS)timing.macros \
          $(MACHINE_MACROS) $(RANDOM).macros \
          dosxyznrc_user_macros.mortran srcxyznrc.macros \
          $(EGS_SOURCEDIR)transportp.macros \
          $(EGS_SOURCEDIR)pegs4_macros.mortran \
          $(EGS_UTILS)phsp_macros.mortran $(IAEA_PHSP_MACROS)\
          $(USER_CODE).mortran \
          $(EGS_SOURCEDIR)get_inputs.mortran \
          $(EGS_SOURCEDIR)get_media_inputs.mortran $(MACHINE_MORTRAN) \
          srcxyznrc.mortran $(RANDOM).mortran \
          $(EGS_UTILS)nrcaux.mortran \
          $(EGS_SOURCEDIR)egs_utilities.mortran \
          $(EGS_SOURCEDIR)egs_parallel.mortran \
          $(EGS_SOURCEDIR)pegs4_routines.mortran \
          $(EGS_SOURCEDIR)egsnrc.mortran
```

From **$EGS_HOME/dosxyznrc/Makefile**

- MORTRAN creates **dosxyznrc_yourMachine.F**
- Fortran compiles the binary **dosxyznrc**

MORTRAN

egs++

Looking inside egs++ applications

- egs++ is a C++ class library
- Applications inherit the class **EGS_AdvancedApplication**

- Shared classes:
  - **particles sources**
  - **geometries**
  - **input/output** functionality

- Inherit application class, scoring routines, I/O functions, etc.

```
#include "egs_advanced_application.h"
#include "egs_scoring.h"
#include "egs_interface2.h"
#include "egs_functions.h"
#include "egs_input.h"
#include "egs_base_source.h"
#include "egs_rndm.h"

class APP_EXPORT Tutor7_Application : public EGS_AdvancedApplication {
     ...
```

SOURCE

class **EGS_BaseSource**
Base source class. All particle sources must be de

class **EGS_BaseSpectrum**
Base class for energy spectra. All energy spectra i

class **EGS_BaseSimpleSource**
Base class for 'simple' particle sources. **More...**

class **EGS_AngularSpreadSource**
A source that adds additional Gaussian angular sp

class **EGS_BeamSource**
A BEAM simulation source. **More...**

class **EGS_CollimatedSource**
A collimated particle source. **More...**

class **EGS_FanoSource**
A Fano source. **More...**

class **EGS_IsotropicSource**
An isotropic source. **More...**

class **EGS_ParallelBeam**
A parallel beam. **More...**

class **EGS_PhspSource**
A phase-space file source. **More...**

class **EGS_PointSource**
A point source. **More...**

class **EGS_RadionuclideSource**
A radionuclide source. **More...**

class **EGS_SourceCollection**
A source collection. **More...**

class **EGS_TransformedSource**
A transformed source. **More...**

class **IAEA_PhspSource**
An IAEA phase-space file source. **More...**

GEOMETRY

class **EGS_Box**
A box geometry. **More...**

class **EGS_SimpleCone**
A single cone that may be open (i.e. extends to infinity

class **EGS_ParallelCones**
A set of "parallel cones" (*i.e.* cones with the same axis

class **EGS_ConeSet**
A set of cones with different opening angles but the sa

class **EGS_ConeStack**
A cone stack. **More...**

class **EGS_CylindersT< T >**
A set of concentric cylinders. **More...**

class **EGS_IPlanes**
A set of planes intersecting in the same axis. **More...**

class **EGS_PlanesT< T >**
A set of parallel planes. **More...**

class **EGS_PlaneCollection**
A collection of non-parallel planes. **More...**

class **EGS_PrismT< T >**
A class for modeling prisms. **More...**

class **EGS_PyramidT< T >**
A template class for modeling pyramids. **More...**

class **EGS_RoundRectCylindersT< Tx, Ty >**
A set of concentric rounded rectangles. **More...**

class **EGS_Space**
The entire space as a geometry object. **More...**

class **EGS_cSpheres**
A set of concentric spheres. **More...**

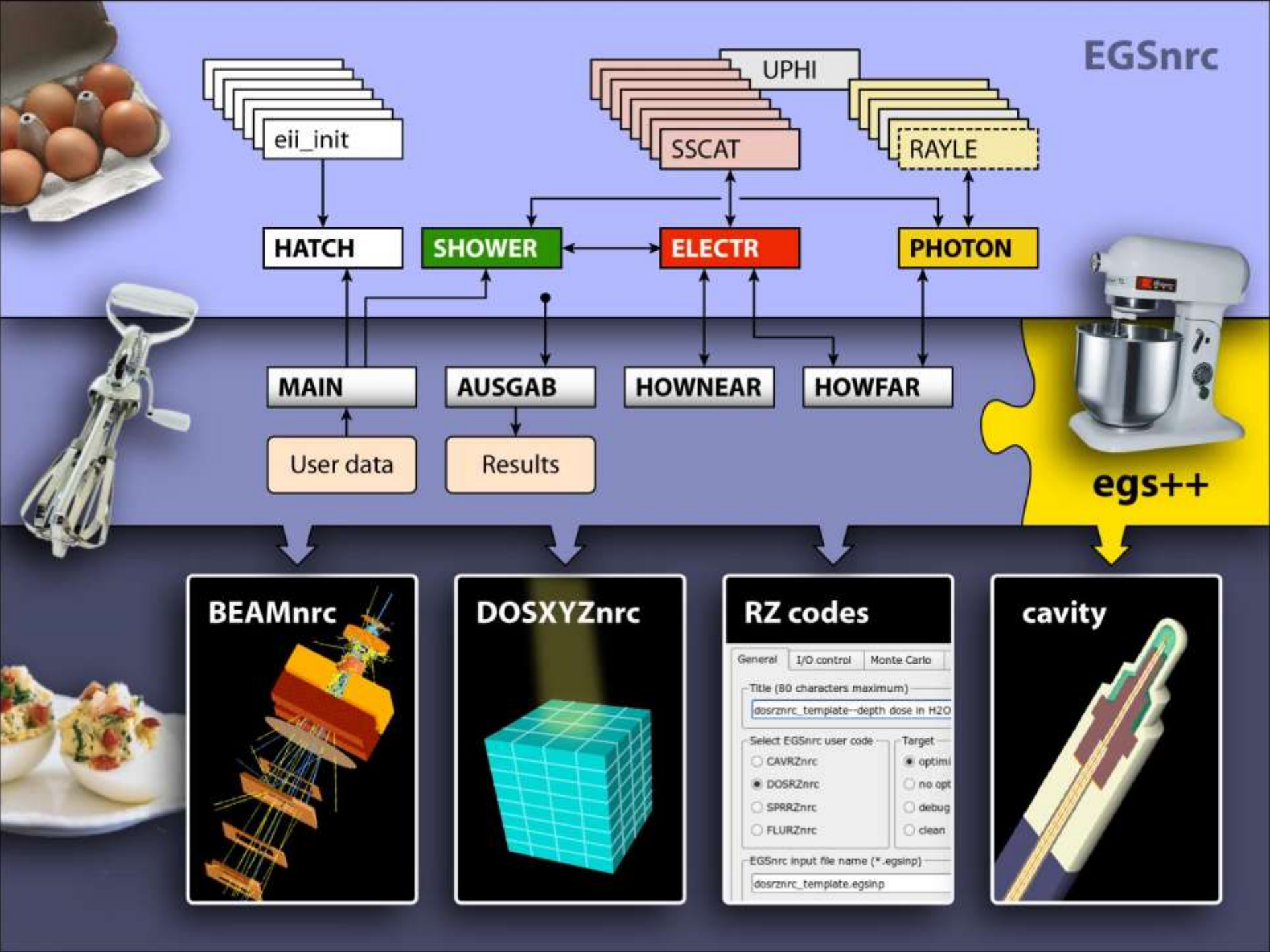
egs++ interfaces with EGSnrc core

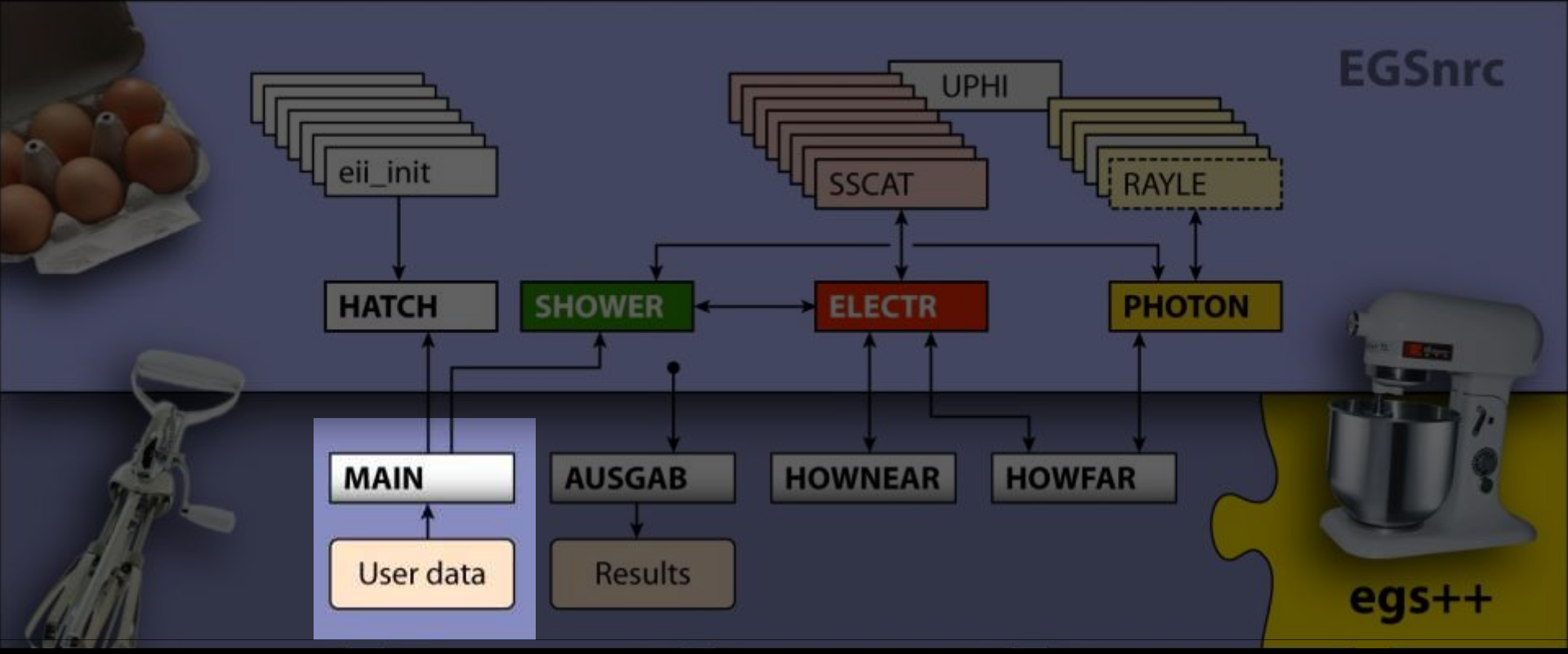
- Most physics still occurs in EGSnrc core
- Functions are mapped between C++/MORTRAN

```
#define egsHatch F77_OBJ_(egs_hatch,EGS_HATCH)
extern __extc__ void egsHatch(void);
```

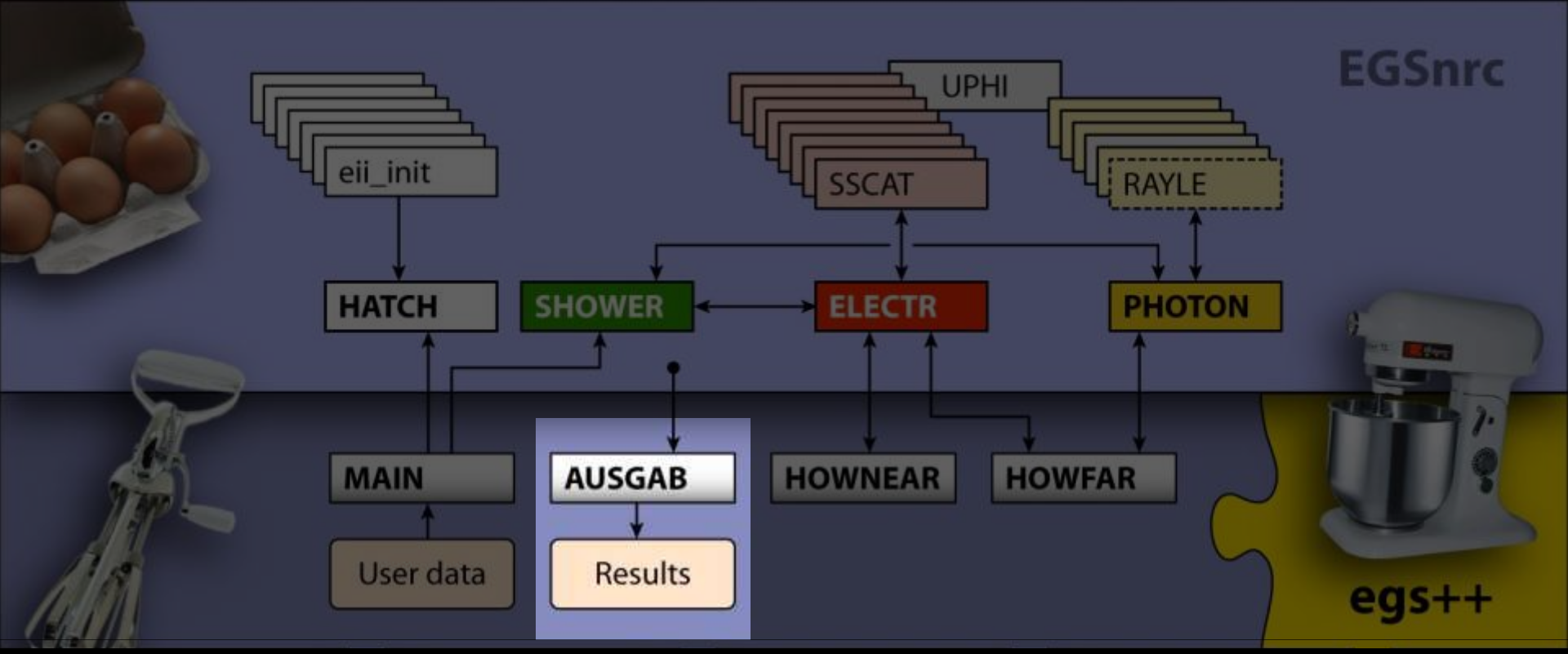
egs++

Application functions

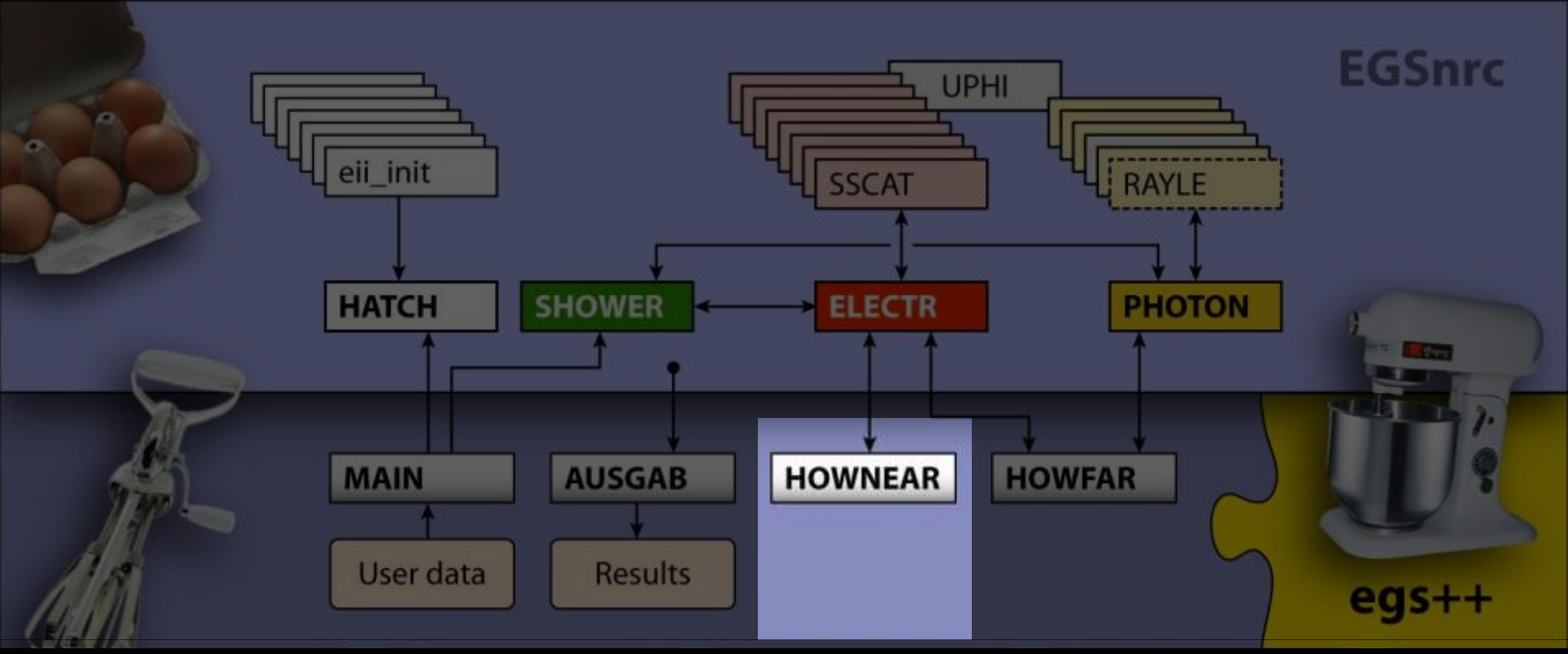

EGSnrc
eii_init
UPHI
SSCAT
RAYLE
HATCH
SHOWER
ELECTR
PHOTON
MAIN
AUSGAB
HOWNEAR
HOWFAR
User data
Results
egs++
BEAMnrc
DOSXYZnrc
RZ codes
General
I/O control
Monte Carlo
Title (80 characters maximum)
dosrznrc_template--depth dose in H2O
Select EGSnrc user code
CAVRZnrc
DOSRZnrc
SPRRZnrc
FLURZnrc
Target
optimi
no opt
debug
clean
EGSnrc input file name (*.egsinp)
dosrznrc_template.egsinp
cavity



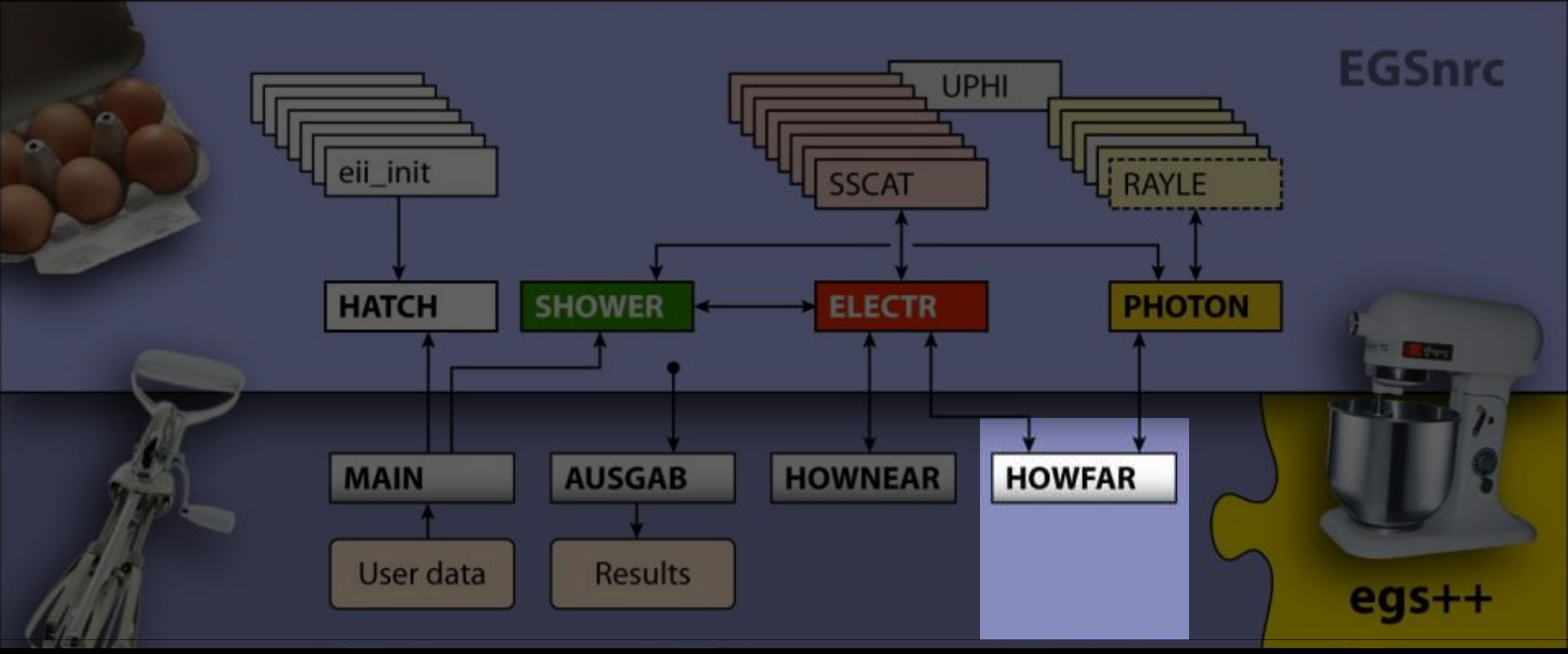
- Read input files
- Initialize/finalize
- Get source particles
- Ask for transport



- Output control
- Receives “call-backs” during transport
  - E.g. after every Compton event
  - Scores results



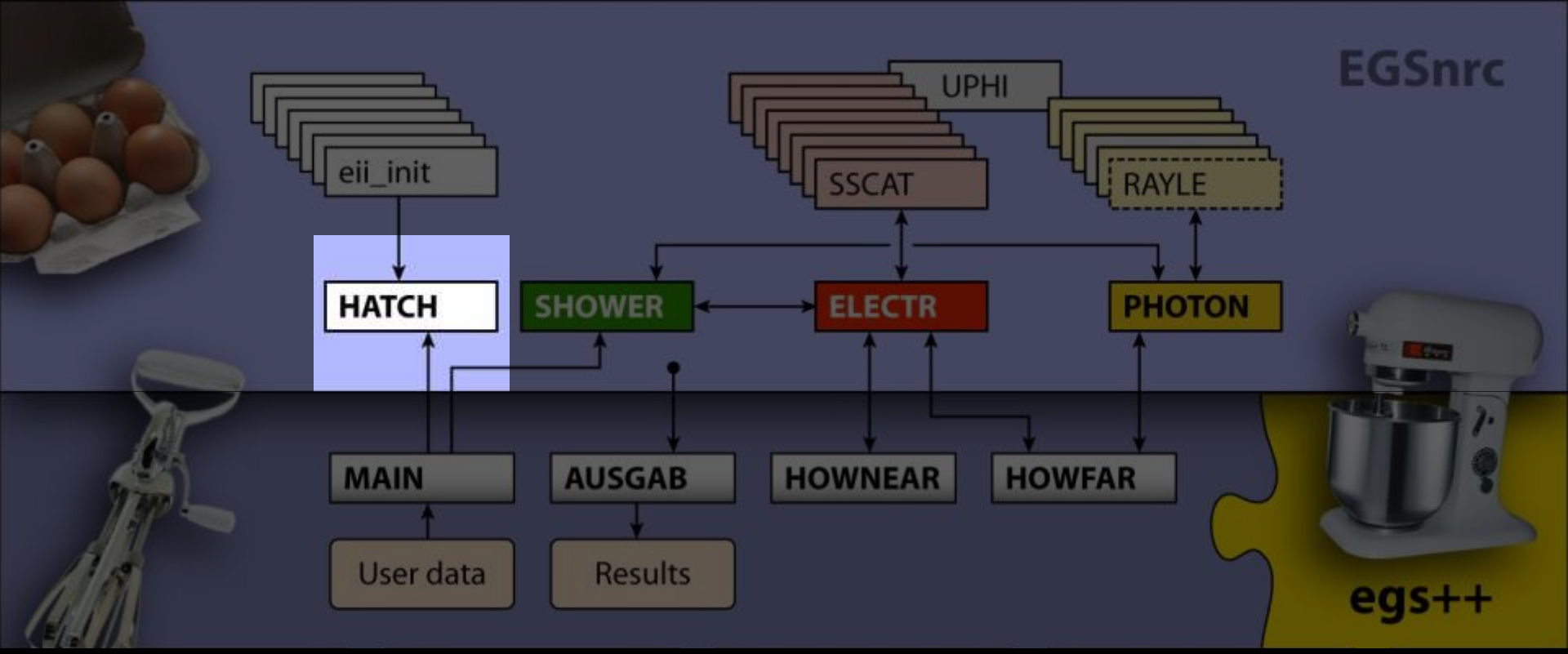
- Defined by geometry
- Calculates perpendicular distance to nearest region boundary
  - Essential for charged particle transport



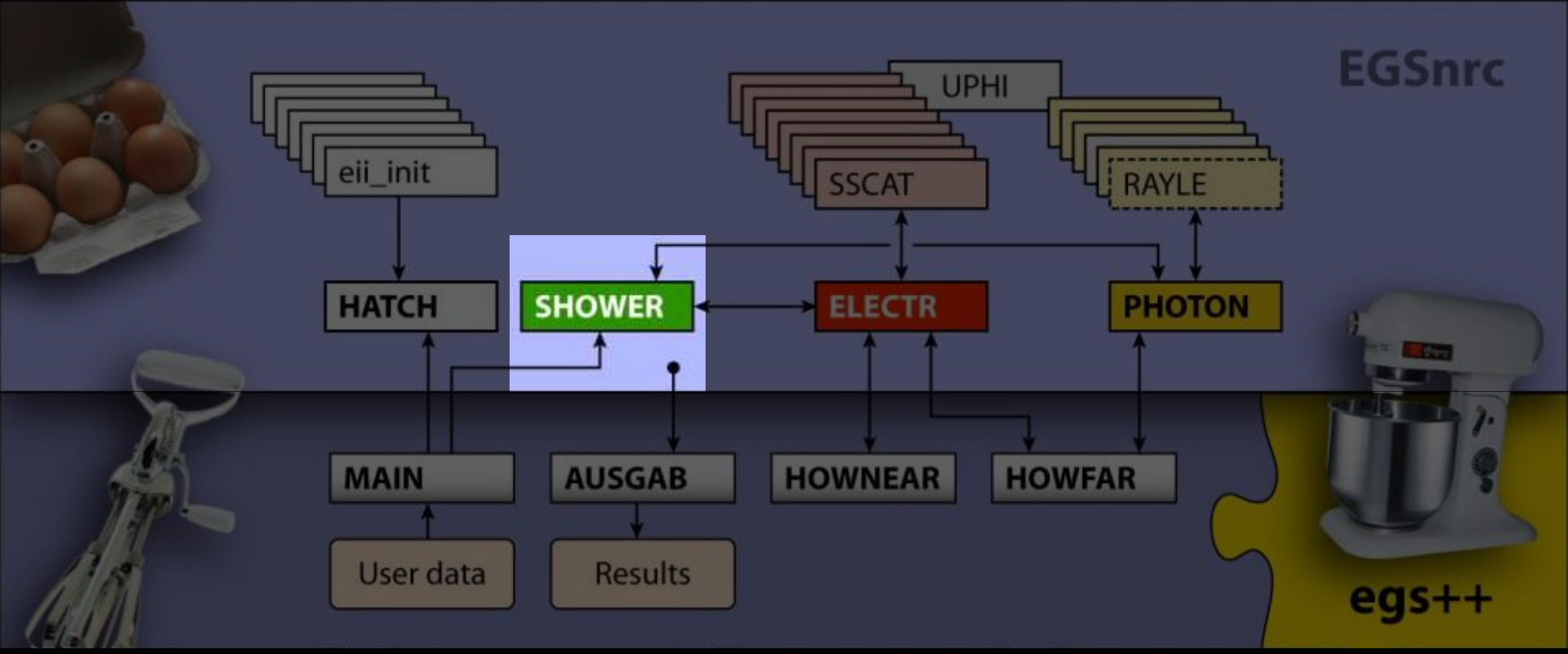
- Defined by geometry
- Calculates distance to the next boundary in the particle's direction
  - If intended particle step USTEP is larger than this distance, resets USTEP to the distance to the boundary
  - If particles reaches a boundary, resets the current region index to the new region

Application functions

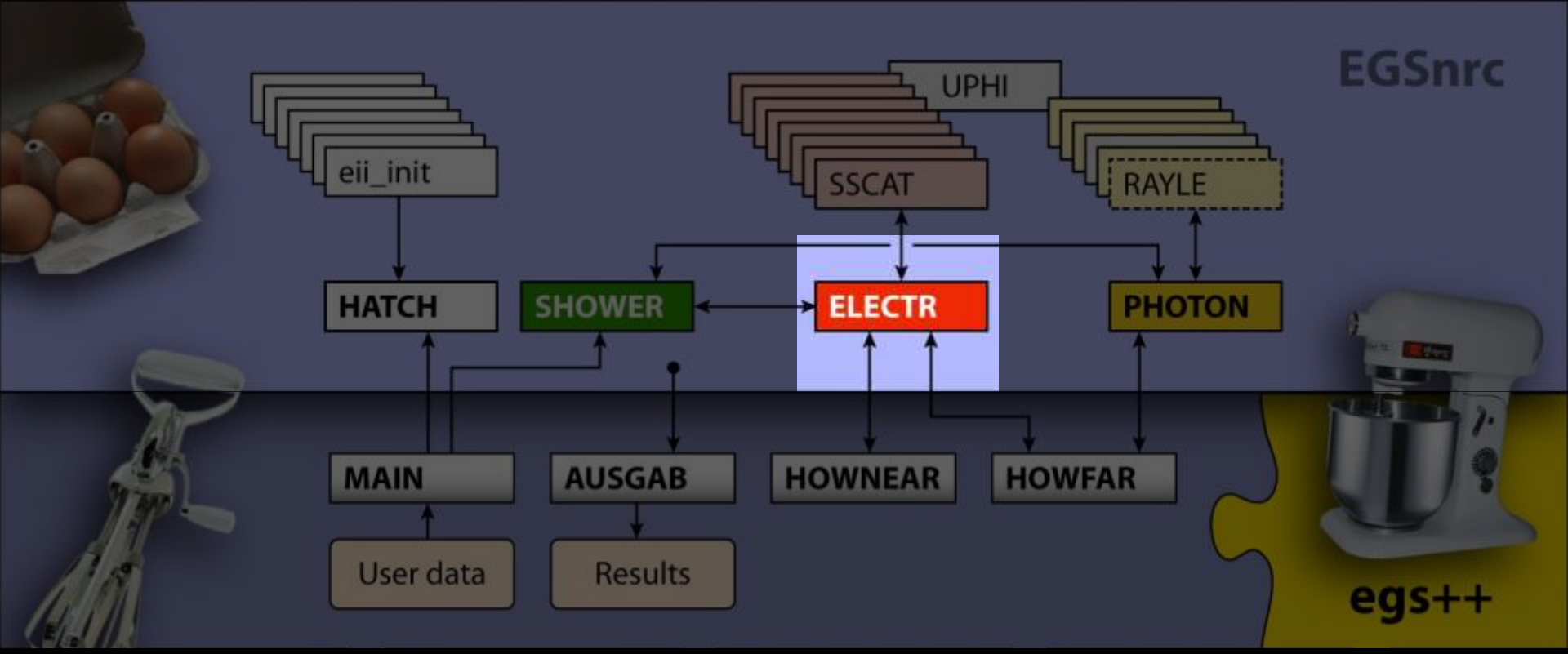
EGSnrc core



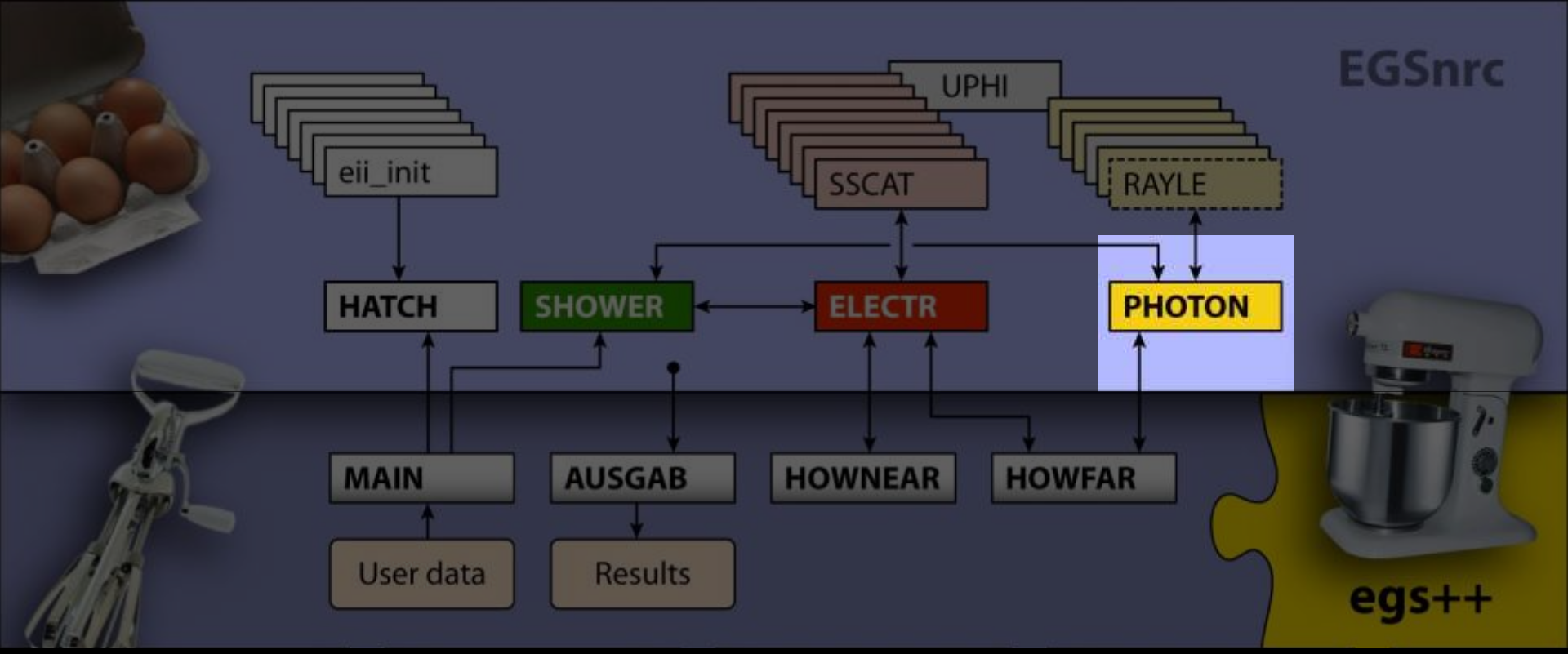
- Initialization
  - Cross section data
  - Atomic relaxation data
  - Material data (PEGS4, density corrections)



- Starts transport from a source particle
- Also starts transport of any secondaries
- Essentially just calls **ELECTR()** or **PHOTON()** in a loop



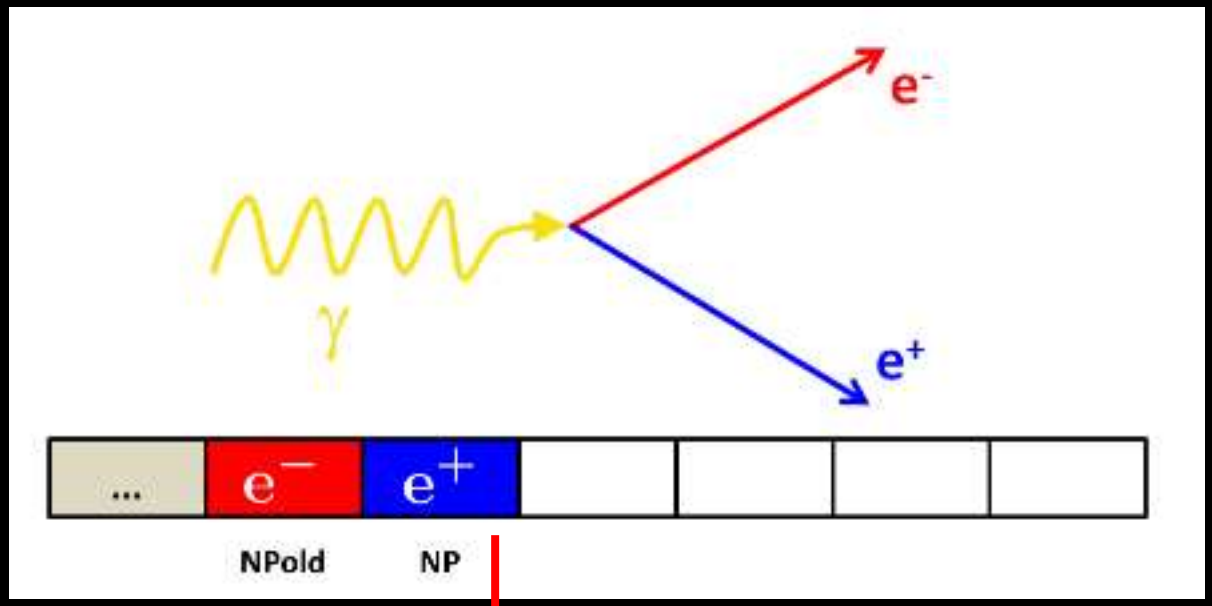
- Transports an electron until it is absorbed or escapes
- Adds secondary particles to the **stack**



- Transports a photon until it is absorbed or escapes
- Adds secondary particles to the **stack**

The stack is an array of particles

- Secondary particles get placed on the stack *for later*
- Ausgab/scoring routines often analyze the stack



particle stack

NPold	NP	IR(NP)	$\vec{X}$(3,NP)	$\vec{\Omega}$(3,NP)	E(NP)	IQ(NP)	WT(NP)	DNEAR(NP)	LATCH(NP)

EGSnrc core

File structure

Your input files go in $EGS_HOME

- You set* the $EGS_HOME environment variable, e.g.

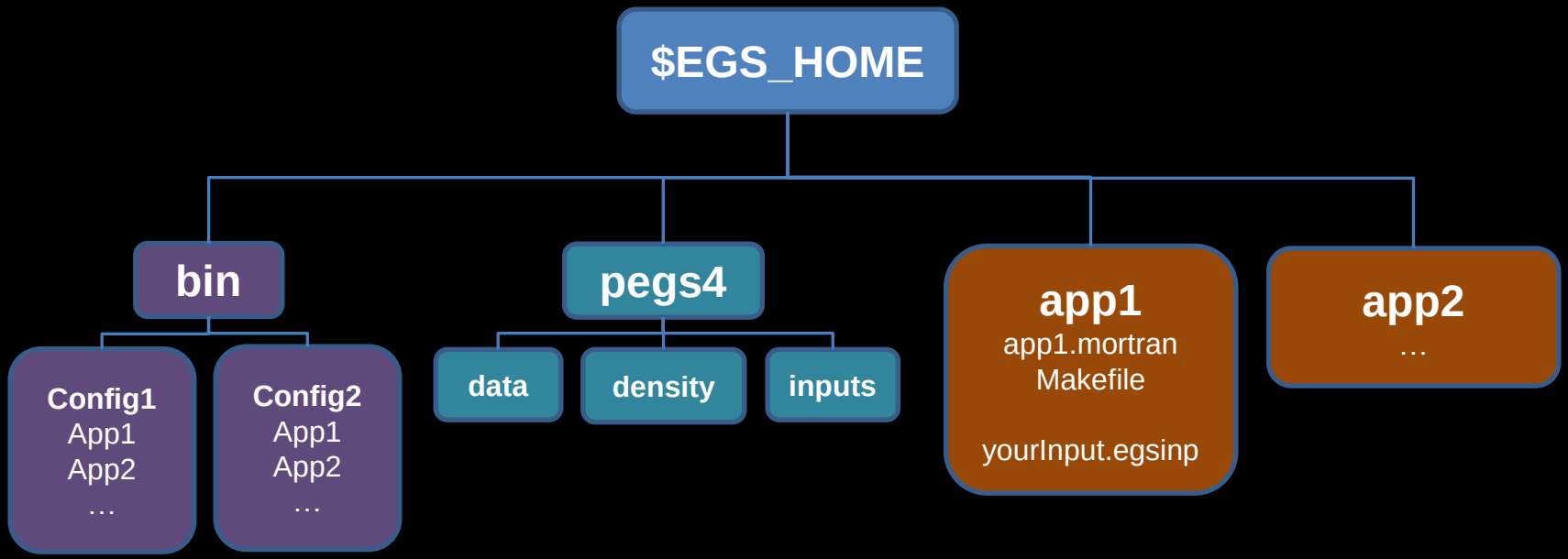
  EGS_HOME = /home/you/EGSnrc/egs_home

- Each application has a folder in $EGS_HOME, e.g.

  $EGS_HOME/dosxyznrc

- Also contains:
  - input files
  - custom data (PEGS4)
  - compiled application binaries

***No spaces or special characters in file paths!**
Keep paths *short


$EGS_HOME
bin
Config1
App1
App2
…
Config2
App1
App2
…
pegs4
data
density
inputs
app1
app1.mortran
Makefile
yourInput.egsinp
app2
…

The EGS are in the $HEN_HOUSE

- You set the $HEN_HOUSE environment variable, e.g.

  HEN_HOUSE = /home/you/EGSnrc/HEN_HOUSE

- Contains code & data
  - EGSnrc core
  - egs++ core
  - Data files
  - Installation & other tools

$HEN_HOUSE

Category	Folder	Description
EGSnrc core	omega	- BEAMnrc source code - addphsp, beamdp, ctcreate, dosxyz_show, readphsp, and statdose - tcl GUIs for beamdp, BEAMnrc and DOSXYZnrc
	previewRZ	- tcl GUI for viewing geometries from DOSRZnrc, CAVRZnrc, FLURZnrc and SPRRZnrc
	src	- EGSnrc core MORTRAN source code
	user_codes	- Application source code (C++ and MORTRAN)
	utils	- Various MORTRAN tools
egs++ and Qt	cutils	- C tools for parallel jobs and BEAMnrc sources
	egs++	- C++ class library - ausgab objects, geometries and sources
	gui	- Qt GUIs: egs_configure, egs_gui and egs_inprz
	iaea_phsp	- C++ tools for IAEA phase-spaces
	interface	- The code that connects C++ and MORTRAN
data	data	- photon cross section data - bremsstrahlung cross section corrections - eii cross sections - molecular form factors
	pegs4	- PEGS4 mortran code - PEGS4 data files - density correction files
	spectra	- various energy spectrum data - radionuclide decay data
configuration	makefiles	- template application makefiles
	mortran3	- the MORTRAN core code
	pieces	- system test code
	scripts	- configuration and support scripts
	specs	- configuration specifications
documentation	docs	- documentation source code

Category	Folder	Description
EGSnrc core	omega	- BEAMnrc source code - addphsp, beamdp, ctcreate, dosxyz_show, readphsp, and statdose - tcl GUIs for beamdp, BEAMnrc and DOSXYZnrc
	previewRZ	- tcl GUI for viewing geometries from DOSRZnrc, CAVRZnrc, FLURZnrc and SPRRZnrc
	src	- EGSnrc core MORTRAN source code
	user_codes	- Application source code (C++ and MORTRAN)
	utils	- Various MORTRAN tools
egs++ and Qt	cutils	- C tools for parallel jobs and BEAMnrc sources
	egs++	- C++ class library - ausgab objects, geometries and sources
	gui	- Qt GUIs: egs_configure, egs_gui and egs_inprz
	iaea_phsp	- C++ tools for IAEA phase-spaces
	interface	- The code that connects C++ and MORTRAN
data	data	- photon cross section data - bremsstrahlung cross section corrections - eii cross sections - molecular form factors
	pegs4	- PEGS4 mortran code - PEGS4 data files - density correction files
	spectra	- various energy spectrum data - radionuclide decay data
configuration	makefiles	- template application makefiles
	mortran3	- the MORTRAN core code
	pieces	- system test code
	scripts	- configuration and support scripts
	specs	- configuration specifications
documentation	docs	- documentation source code

$HEN_HOUSE

Category	Folder	Description
EGSnrc core	omega	- BEAMnrc source code - addphsp, beamdp, ctcreate, dosxyz_show, readphsp, and statdose - tcl GUIs for beamdp, BEAMnrc and DOSXYZnrc
	previewRZ	- tcl GUI for viewing geometries from DOSRZnrc, CAVRZnrc, FLURZnrc and SPRRZnrc
	src	- EGSnrc core MORTRAN source code
	user_codes	- Application source code (C++ and MORTRAN)
	utils	- Various MORTRAN tools
egs++ and Qt	cutils	- C tools for parallel jobs and BEAMnrc sources
	egs++	- C++ class library - ausgab objects, geometries and sources
	gui	- Qt GUIs: egs_configure, egs_gui and egs_inprz
	iaea_phsp	- C++ tools for IAEA phase-spaces
	interface	- The code that connects C++ and MORTRAN
data	data	- photon cross section data - bremsstrahlung cross section corrections - eii cross sections - molecular form factors
	pegs4	- PEGS4 mortran code - PEGS4 data files - density correction files
	spectra	- various energy spectrum data - radionuclide decay data
configuration	makefiles	- template application makefiles
	mortran3	- the MORTRAN core code
	pieces	- system test code
	scripts	- configuration and support scripts
	specs	- configuration specifications
documentation	docs	- documentation source code

$HEN_HOUSE

Category	Folder	Description
EGSnrc core	omega	- BEAMnrc source code - addphsp, beamdp, ctcreate, dosxyz_show, readphsp, and statdose - tcl GUIs for beamdp, BEAMnrc and DOSXYZnrc
	previewRZ	- tcl GUI for viewing geometries from DOSRZnrc, CAVRZnrc, FLURZnrc and SPRRZnrc
	src	- EGSnrc core MORTRAN source code
	user_codes	- Application source code (C++ and MORTRAN)
	utils	- Various MORTRAN tools
egs++ and Qt	cutils	- C tools for parallel jobs and BEAMnrc sources
	egs++	- C++ class library - ausgab objects, geometries and sources
	gui	- Qt GUIs: egs_configure, egs_gui and egs_inprz
	iaea_phsp	- C++ tools for IAEA phase-spaces
	interface	- The code that connects C++ and MORTRAN
data	data	- photon cross section data - bremsstrahlung cross section corrections - eii cross sections - molecular form factors
	pegs4	- PEGS4 mortran code - PEGS4 data files - density correction files
	spectra	- various energy spectrum data - radionuclide decay data
configuration	makefiles	- template application makefiles
	mortran3	- the MORTRAN core code
	pieces	- system test code
	scripts	- configuration and support scripts
	specs	- configuration specifications
documentation	docs	- documentation source code

$HEN_HOUSE

Category	Folder	Description
EGSnrc core	omega	- BEAMnrc source code - addphsp, beamdp, ctcreate, dosxyz_show, readphsp, and statdose - tcl GUIs for beamdp, BEAMnrc and DOSXYZnrc
	previewRZ	- tcl GUI for viewing geometries from DOSRZnrc, CAVRZnrc, FLURZnrc and SPRRZnrc
	src	- EGSnrc core MORTRAN source code
	user_codes	- Application source code (C++ and MORTRAN)
	utils	- Various MORTRAN tools
egs++ and Qt	cutils	- C tools for parallel jobs and BEAMnrc sources
	egs++	- C++ class library - ausgab objects, geometries and sources
	gui	- Qt GUIs: egs_configure, egs_gui and egs_inprz
	iaea_phsp	- C++ tools for IAEA phase-spaces
	interface	- The code that connects C++ and MORTRAN
data	data	- photon cross section data - bremsstrahlung cross section corrections - eii cross sections - molecular form factors
	pegs4	- PEGS4 mortran code - PEGS4 data files - density correction files
	spectra	- various energy spectrum data - radionuclide decay data
configuration	makefiles	- template application makefiles
	mortran3	- the MORTRAN core code
	pieces	- system test code
	scripts	- configuration and support scripts
	specs	- configuration specifications
documentation	docs	- documentation source code

$HEN_HOUSE

Category	Folder	Description
EGSnrc core	omega	- BEAMnrc source code - addphsp, beamdp, ctcreate, dosxyz_show, readphsp, and statdose - tcl GUIs for beamdp, BEAMnrc and DOSXYZnrc
	previewRZ	- tcl GUI for viewing geometries from DOSRZnrc, CAVRZnrc, FLURZnrc and SPRRZnrc
	src	- EGSnrc core MORTRAN source code
	user_codes	- Application source code (C++ and MORTRAN)
	utils	- Various MORTRAN tools
egs++ and Qt	cutils	- C tools for parallel jobs and BEAMnrc sources
	egs++	- C++ class library - ausgab objects, geometries and sources
	gui	- Qt GUIs: egs_configure, egs_gui and egs_inprz
	iaea_phsp	- C++ tools for IAEA phase-spaces
	interface	- The code that connects C++ and MORTRAN
data	data	- photon cross section data - bremsstrahlung cross section corrections - eii cross sections - molecular form factors
	pegs4	- PEGS4 mortran code - PEGS4 data files - density correction files
	spectra	- various energy spectrum data - radionuclide decay data
configuration	makefiles	- template application makefiles
	mortran3	- the MORTRAN core code
	pieces	- system test code
	scripts	- configuration and support scripts
	specs	- configuration specifications
documentation	docs	- documentation source code

File structure

Simulations

Compile on the command line

- Compile applications using ‘make’
- On linux:

```
cd $EGS_HOME/dosxyznrc
make clean; make
```

- On Windows (with mingw compilers):

```
mingw32-make clean
mingw32-make
```

- Input files must be in the **application folder** in $EGS_HOME

Execute on the command line

- To run a simulation on a single core (e.g. egs_app):

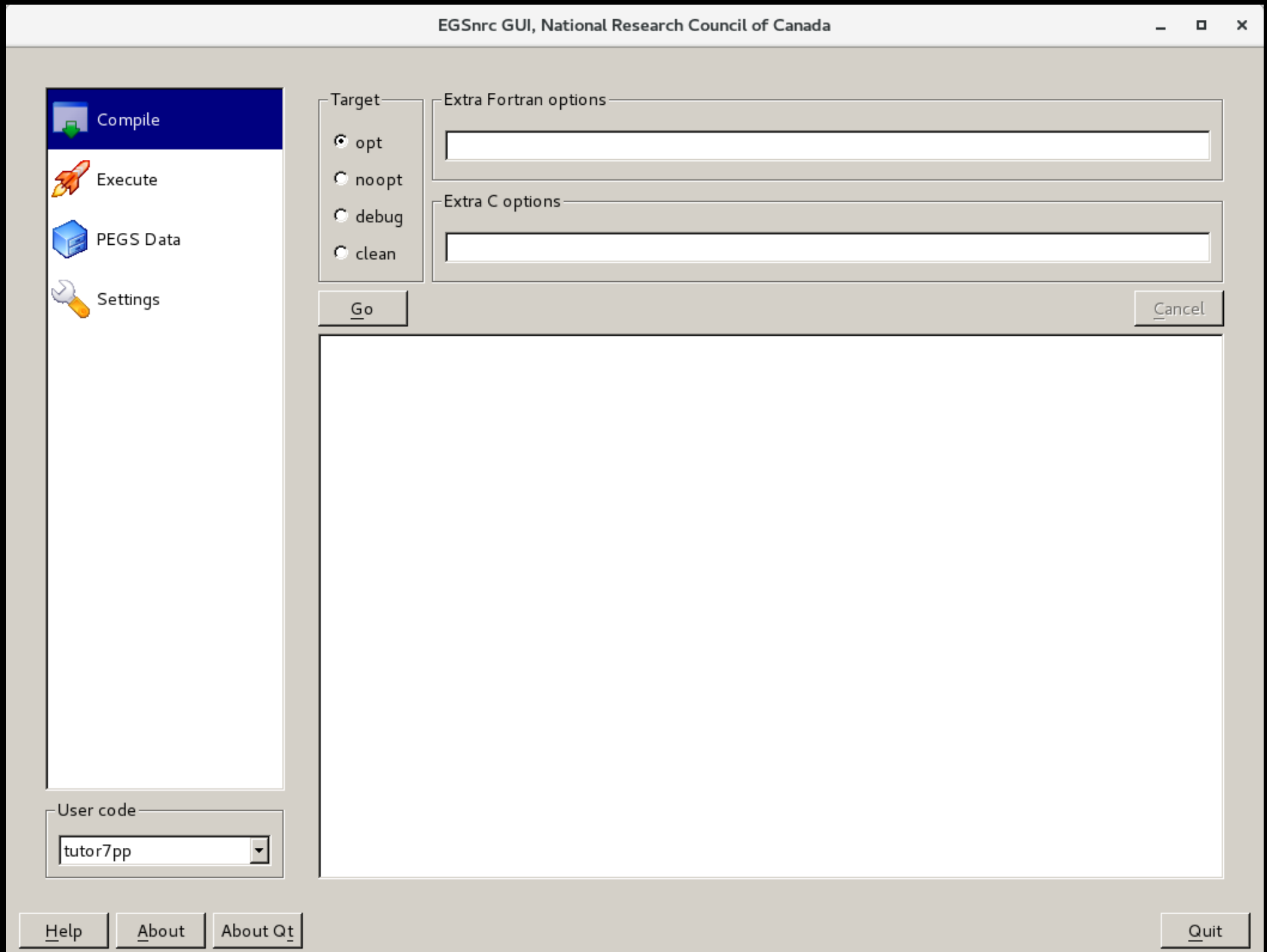
```
egs_app -i myInput [-p myPegs] [other_args]
```

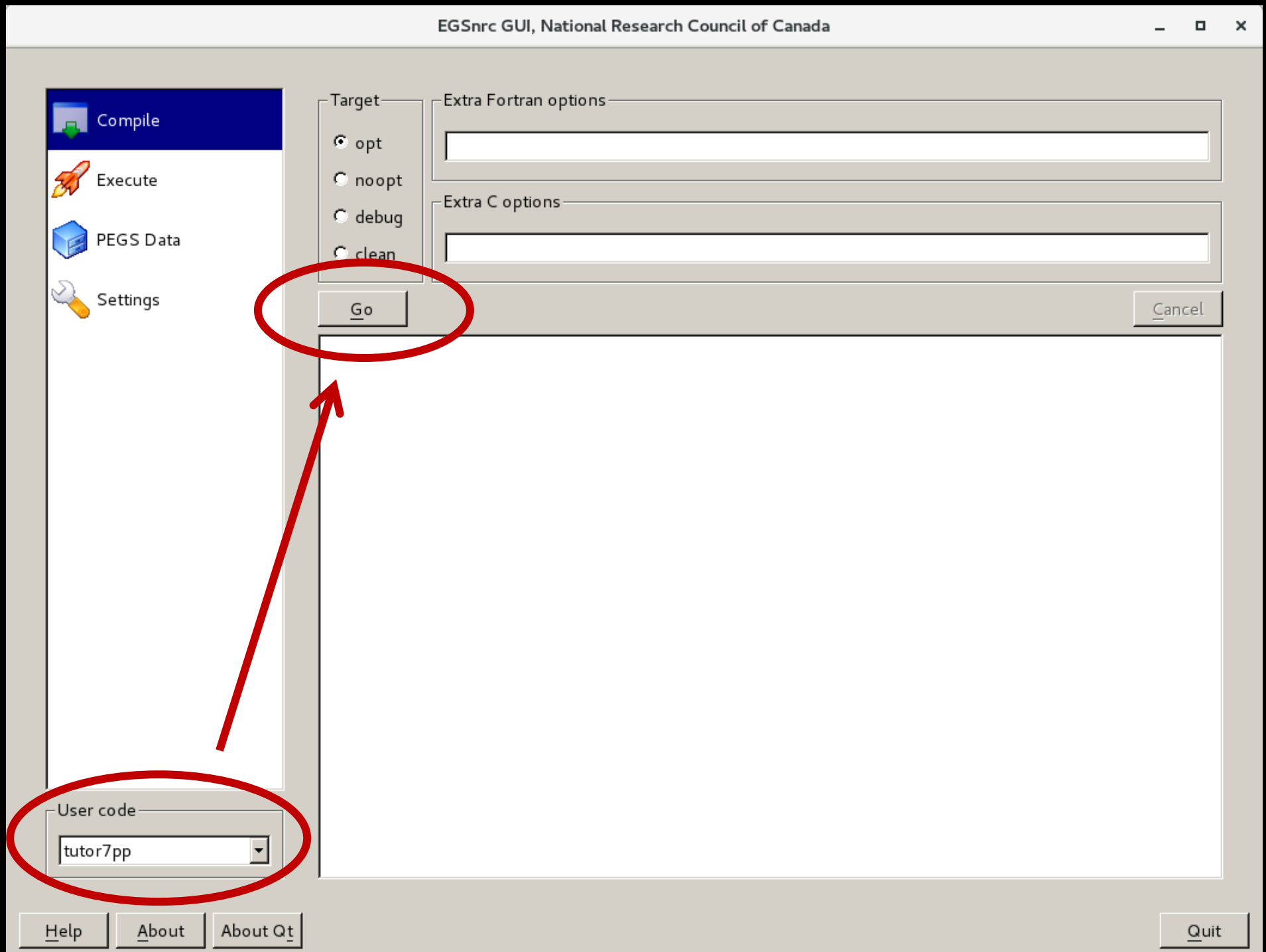
- To run in parallel on local cores (no queueing system):

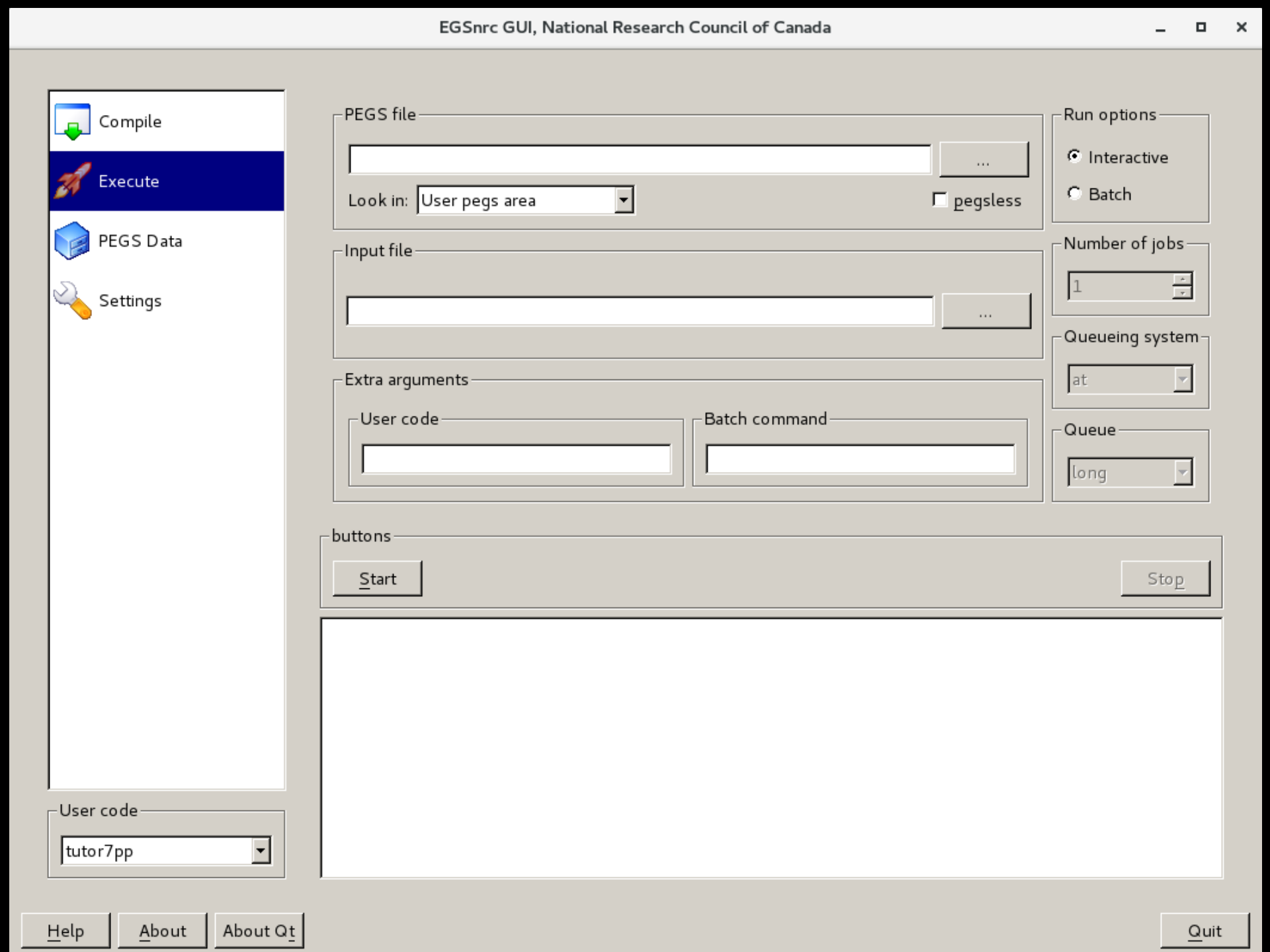
```
egs-parallel -c 'egs_app -i myInput' --batch cpu -n4 -v
```

- For pbs queueing system:

```
egs-parallel -c 'egs_app -i myInput' -q user1 --batch
pbsdsh -n50 -v
```


EGSnrc GUI, National Research Council of Canada
Compile
Execute
PEGS Data
Settings
User code
tutor7pp
Target
opt
noopt
debug
clean
Extra Fortran options
Extra C options
Go
Cancel
Help
About
About Qt
Quit


EGSnrc GUI, National Research Council of Canada
Compile
Execute
PEGS Data
Settings
Target
opt
noopt
debug
clean
Extra Fortran options
Extra C options
Go
Cancel
User code
tutor7pp
Help
About
About Qt
Quit


EGSnrc GUI, National Research Council of Canada
Compile
Execute
PEGS Data
Settings
User code
tutor7pp
PEGS file
...
Look in:
User pegs area
pegsless
Input file
...
Extra arguments
User code
Batch command
Run options
Interactive
Batch
Number of jobs
1
Queueing system
at
Queue
long
buttons
Start
Stop
Help
About
About Qt
Quit