

INTERFACING ROXIE

MATTHIAS BONORA



MT-28
International Conference
on Magnet Technology

Aix-en-Provence, France, 10-15 September 2023



MT-28
International Conference
on Magnet Technology
Aix-en-Provence, France, 10-15 September 2023

Overview

- Interfaces from ROXIE
 - xml, hmo, geometry
 - Interfacing during execution
- Scripting ROXIE
 - Python API, interfacing API
- Working with Systems and Models (MBSE)
 - pyMBSE, MMBSE, developments



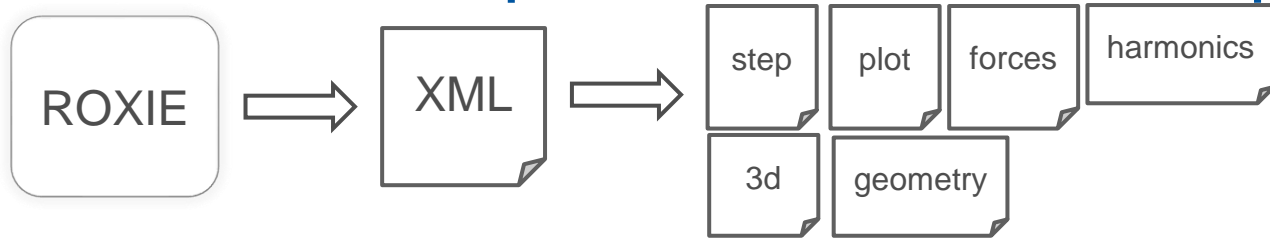
Interfaces from ROXIE

- External mesh input (.hmo)
- Simplify ROXIE output structure
- ~~.output, .post, .force2D, .map2d, .wrl~~
- -> 2 main outputs
 - .pdf for Report
 - .xml for structured output
 - Contains plots, graphs, geometry, dv, objectives, ...

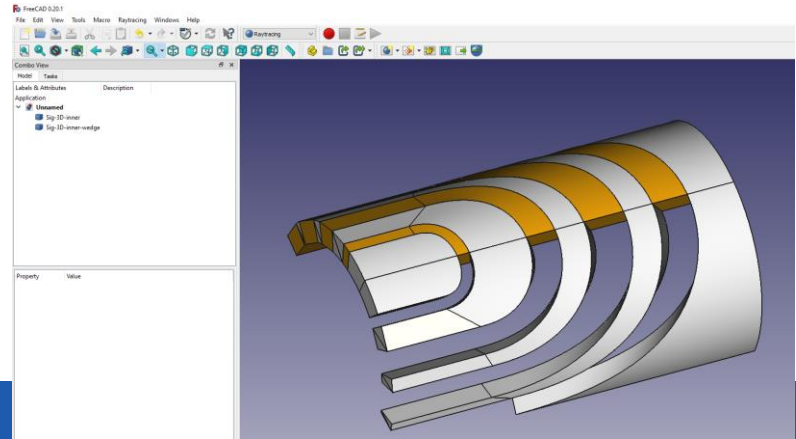
```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE roxieData SYSTEM "roxieData.dtd">
3 <roxieData comment="STATIC 14.5kA" version="23.6.0.1">
4   <plottingInfos>
5     <pageXsec id="1">
6       <coilPlot id="3" level="1"/>
7       <title colour="black" label="STATIC 14.5kA"/>
8       <axisX visible="true" min="-100.000000" max="":
9       <axisY visible="false" min="-1.000000000000e2"
10    </pageXsec>
11    <pageXsec id="2">
12      <meshPlot id="72" level="1"/>
13      <meshPlot id="34" level="2"/>
14      <matrixPlot id="2" level="3"/>
15      <title colour="black" label="STATIC 14.5kA"/>
16      <axisX visible="true" min="-300.000000" max="":
17      <axisY visible="false" min="-3.000000000000e2"
18    </pageXsec>
19    <pageXsec id="3">
20      <meshPlot id="72" level="1"/>
21      <meshPlot id="32" level="2"/>
22      <title colour="black" label="STATIC 14.5kA"/>
23      <axisX visible="true" min="-300.000000" max="":
24      <axisY visible="false" min="-3.000000000000e2"
25    </pageXsec>
```

Interfaces from ROXIE

- Extension to specific format via scripts

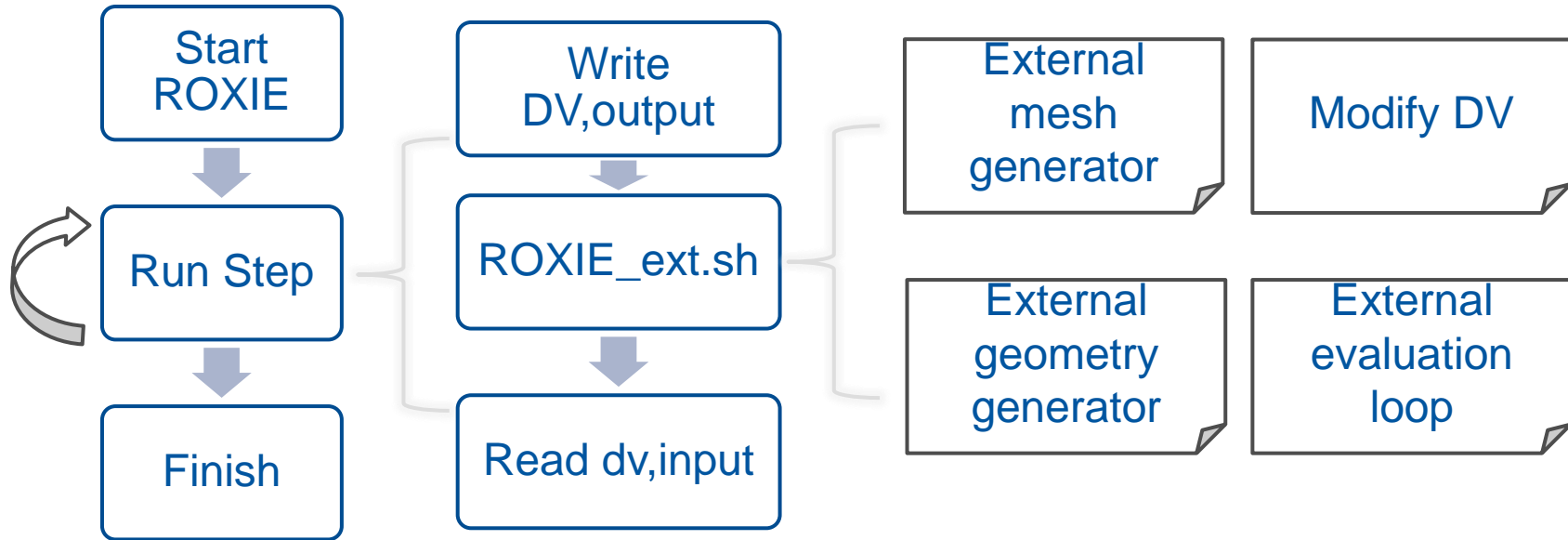


- Standards for output format



Interfaces from ROXIE

- Interfacing during execution (WIP)



Scripting ROXIE

- Extend ROXIE functionality in Python
- ROXIE-api: interact with ROXIE (execution, xml outputs, plots, input file generation)
- ROXIE-interfaces: generate & transform files to use in ROXIE (.hmo, meshes, geometry)

Scripting ROXIE – ROXIE-api

- Documentation
- Handle .data files
 - Load existing models
 - Transform tables, set flags
 - Generate new data files
- Handle xml output files
 - Plot
 - Extract data
- Execute ROXIE

Example gallery

Below is a gallery of examples for using this package

Output parsing and plotting

Examples using the output parsing and plotting functionality



Roxie execution

Examples running roxie, and modifying input files



Scripting ROXIE – input files

- Load existing files
- Operate on tables / flags
- Copy between Files
- Save as new File

Setting up

Load Datafile into RoxieInputBuilder object

```
import pathlib
import tempfile
from roxieapi.input.builder import RoxieInputBuilder

folder = pathlib.Path("../input_files").absolute()
datafile = "dipole_2d.data"

rib = RoxieInputBuilder.from_datafile(folder / datafile)
```

Change flags and metadata

```
rib.comment = "Python modified datafile"
rib.flags["LDEBUG"] = True
rib.flags["LFORCE2D"] = True

rib.flags
```

Check defined Blocks

```
rib.block
```

Update alpha angle of block no 4

```
# Dataframe index 3 corresponds to Block no 4
rib.block.loc[3, "alpha"] = 23.2

rib.block
```

Generate new datafile

```
tmp_dir = pathlib.Path(tempfile.gettempdir())
tmp_file = tmp_dir / "temp.data"
rib.build(str(tmp_file))

with open(tmp_file, "r") as f:
    print(f.read())
```


Scripting ROXIE - execution

- Simple Upload / execution
- Return of output log, status
- Download of results
- Run locally (terminal)
- OR remotely via REST
 - Server, Docker

Executing roxie

Set up paths, datafile and execute Roxie as a process

```
import pathlib
from roxieapi.tool_adapter.RoxieToolAdapter import TerminalRoxieToolAdapter

folder = pathlib.Path("../input_files").absolute()
datafile = "dipole_2d.data"

runner = TerminalRoxieToolAdapter(input_file=datafile, input_folder_rel_dir=folder)

result = runner.run()

print(f"Run executed with return code {result}")
```

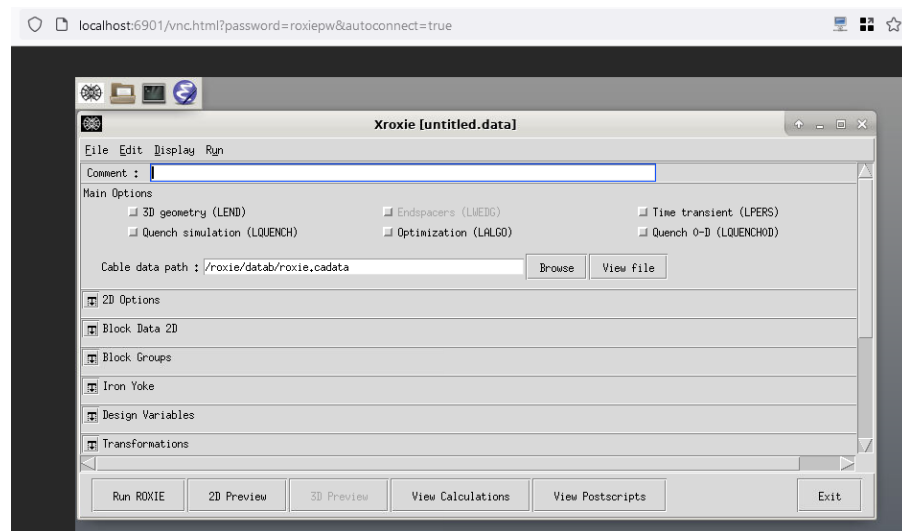
Check output and errors

```
for line in runner.output_lines:
    print(line)
```

```
17 # Executing roxie
18 # -----
19 # Set up paths, datafile and execute Roxie as a process
20 #
21 import pathlib
22 from roxieapi.tool_adapter.RoxieToolAdapter import RestRoxieToolAdapter
23
24 folder = pathlib.Path("../input_files").absolute()
25 datafile = "dipole_2d.data"
26
27 runner = RestRoxieToolAdapter(folder, datafile, "my-model", "localhost", 8080)
28 result = runner.run()
29
30 print(f"Run executed with return code {result}")
31
32 Run Cell | Run Above | Debug Cell
33 # %
34 # Check output and errors
35 # -----
36 |
37 for line in runner.output_lines:
38 |     print(line)
39
```

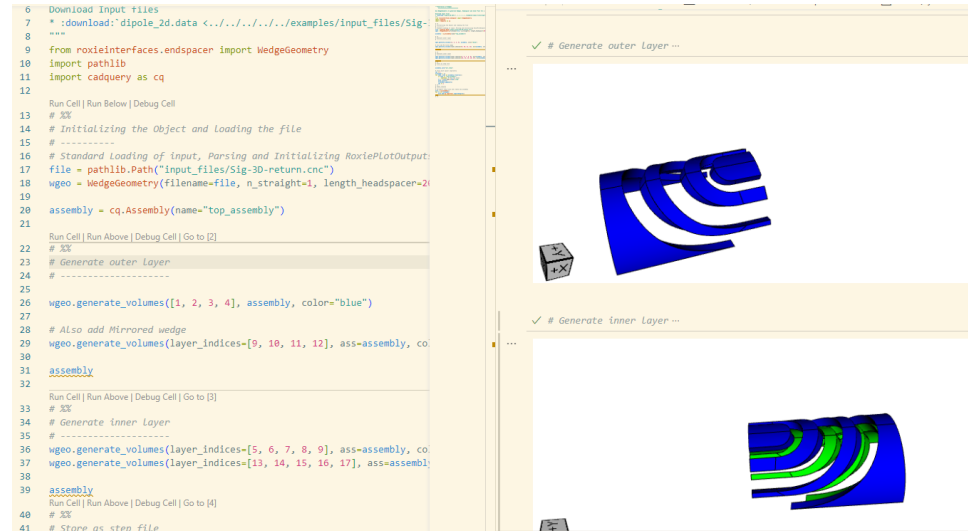
ROXIE in Docker

- ROXIE Docker container (see [doc](#))
 - Without Linux server
 - from any system*
 - VNC Server/client
 - REST API for python



ROXIE Interfaces package

- Documentation
- Collection of scripts for translating IO files
 - ROXIE to STEP (wedges, coils)
 - Hypermesh hmascii to hmo
 - Gmsh to hmo (WIP)



Linking everything together

pyMBSE, MMBSE, Magnet Exchange



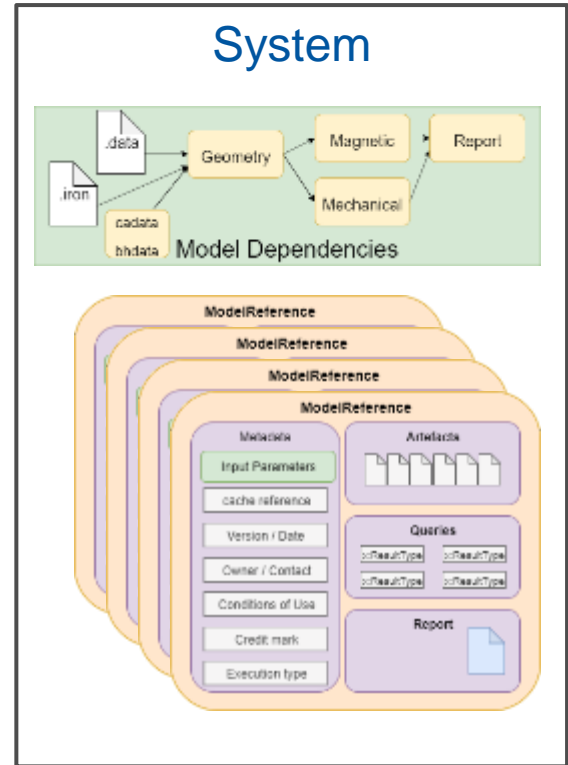
MT-28
International Conference
on Magnet Technology
Aix-en-Provence, France, 10-15 September 2023

2023-09-11

Matthias Bonora

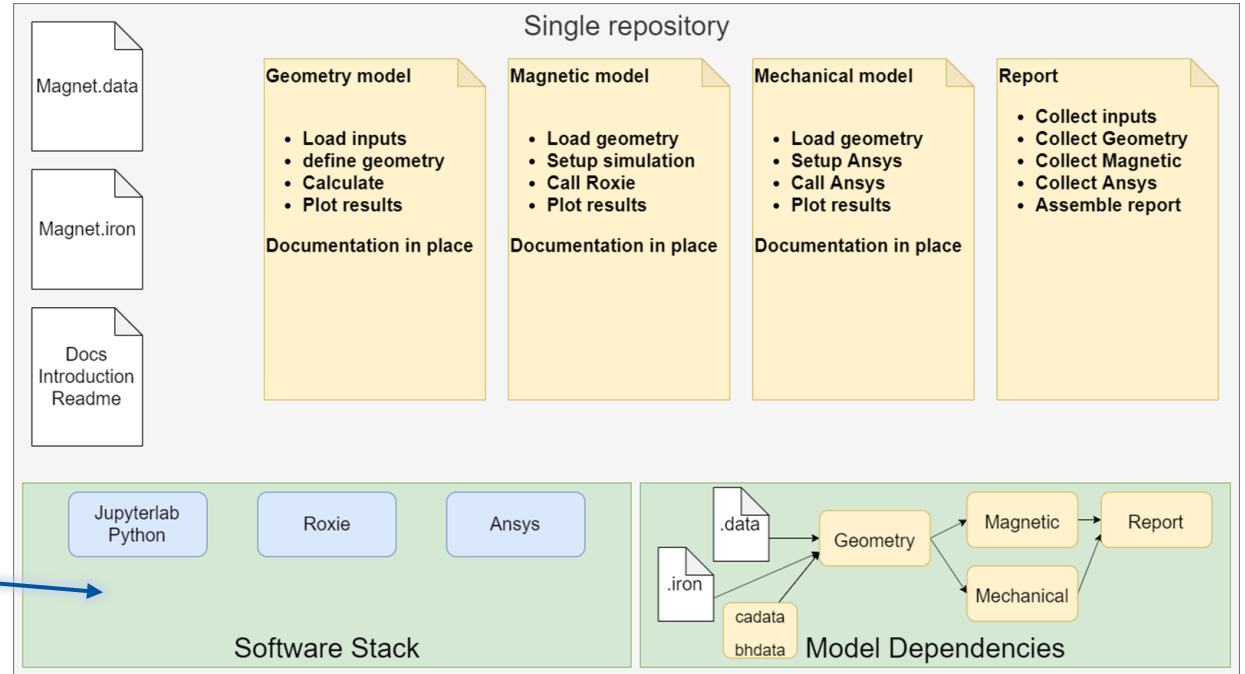
Linking everything together - pyMBSE

- So far: Small modules, packages and projects on top of ROXIE
 - R&D of features and extensions
 - Missing Concept of Integration
- → pyMBSE
 - Started in PSI for the CHART/MagNum project
 - Introduce the concept of models, interfaces and model based system engineering to magnet design
 - Define Magnet Models, dependencies, inputs, outputs, Dependency graph
 - Link with Measurements
 - **Integrated Environment**
 - Software
 - Tools
 - Packages



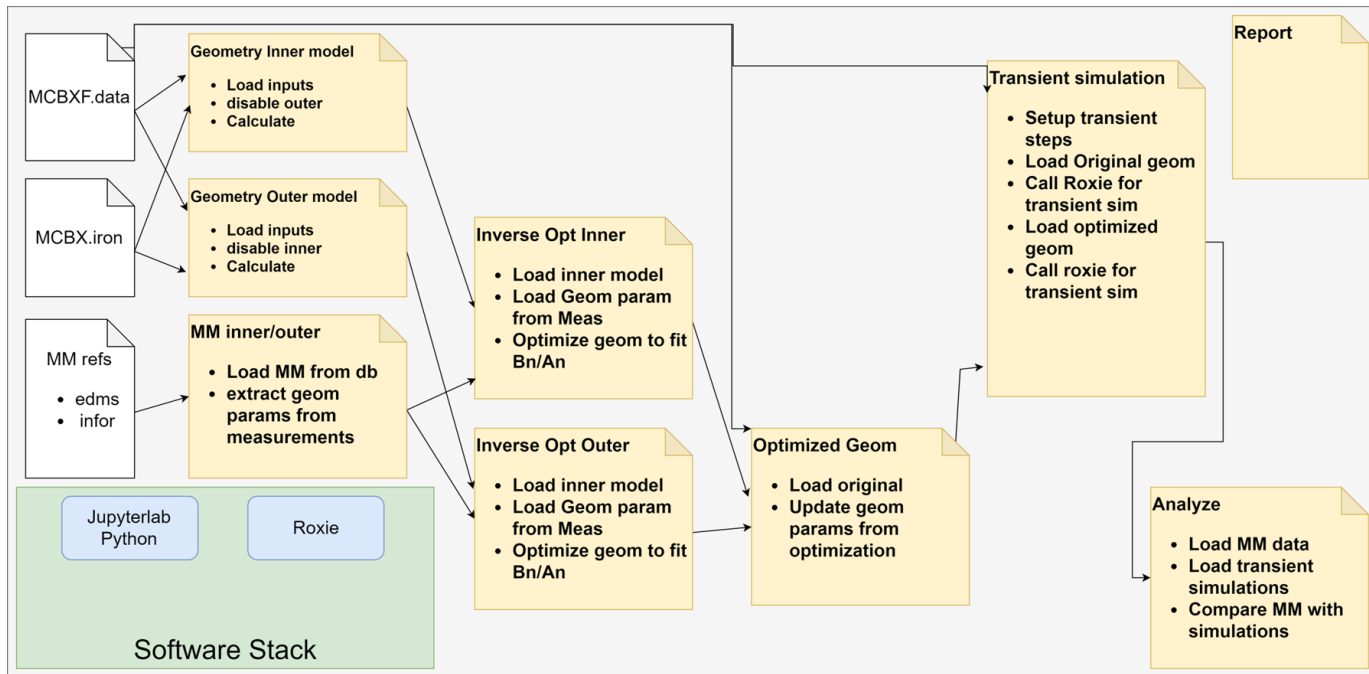
Linking everything together - pyMBSE

Docker
containers



Linking everything together – pyMBSE

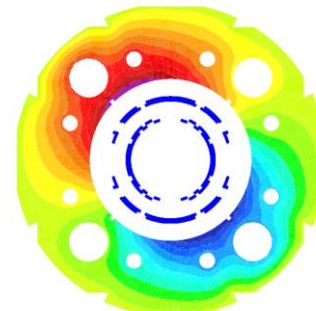
Example: MCBXF



geometrical multipoles only

23/05/23 17:34

A (Tm)



MCBXF report

🔍 Search this book...

MCBXP

Coil Geometry

0. initlab working environment

0. initlab working environment

Optimized Inner Coil Geometry from

measurement data:

Optimized Outer Coil Geometry from

measurement data:

Measurement - Only inner aperture

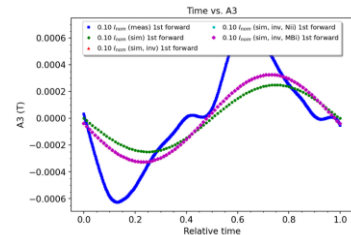
present

PyMagAnalysis - Compare

Simulation with Measurements

```

    Analyzer(references).CircularA(1["C"], angleType=angleType, centered=True).plot()
    Analyzer(inputs).InnerGeom(geomA(1["C"], angleType=angleType).plot())
    Analyzer(inputs).angleResiduals(1["C"], angleType=angleType, references = references, cente
    Analyzer(inputs).angleResiduals(1["C"], angleType=angleType, references = references, cente
  
```



MMBSE

- Database with Magnets and systems
- Different Models
- REST API
 - Scriptable
- Work in Progress
 - Connect to ROXIE GUI
 - Connect with pyMBSE

The screenshot displays the MMBSE web application interface. At the top, the browser address bar shows the URL <https://mmbse.app.cern.ch/systems/41/>. The navigation menu includes 'Home' and 'Systems'. The main content area is titled 'SIGRUM' and contains a 'System Information' section with the following details:

- System Information**
Details, ownership and linked files.
- Project owner**: Stephan Russenschuck
- Description**: Sigrum design study (straight version). Including variants for the coil heads for winding tests.
- Tags**: Dipole, Nb-Ti

Below the system information is a 'System models' section with a 'Create new model' button. It lists models associated with the SIGRUM system:

Type	Name	Design step	Created on	Latest version
ROXIE	SIGRUM Quench Test	quench	Sept. 5, 2023, 12:47 p.m.	View

On the right side of the interface, there are three 3D visualizations of magnet models: a blue grid-based sector, a red and black circular structure, and a yellow and orange curved structure.

MMBSE

- Store Systems
 - Store Inputs
 - Files
 - Dependencies to other Models
 - Parameters
- Store Outputs of execution
 - Version
 - Report
 - Result files
- Access via API (Python)
 - Connect to other System

System Overview

Systems
List with all registered systems.

Type	Name	Created at	
SAMPLE	Building 311 calibration dipole	June 22, 2023, 2:59 p.m.	View
STUDY	SIGRUM	Sept. 4, 2023, 3:02 p.m.	View

2D iron yoke

Model Information
Details, description and linked files.

Type	ROXIE
Part of	SIGRUM
Description	2D iron yoke, sparse mesh, with cooling hole

Inputs

roxie.bhdata	10.6 KB	Download
roxie.cadata	34.5 KB	Download
Sig-2D-iron.data	4.5 KB	Download
Sig-2D-iron.iron	763 bytes	Download

Outputs

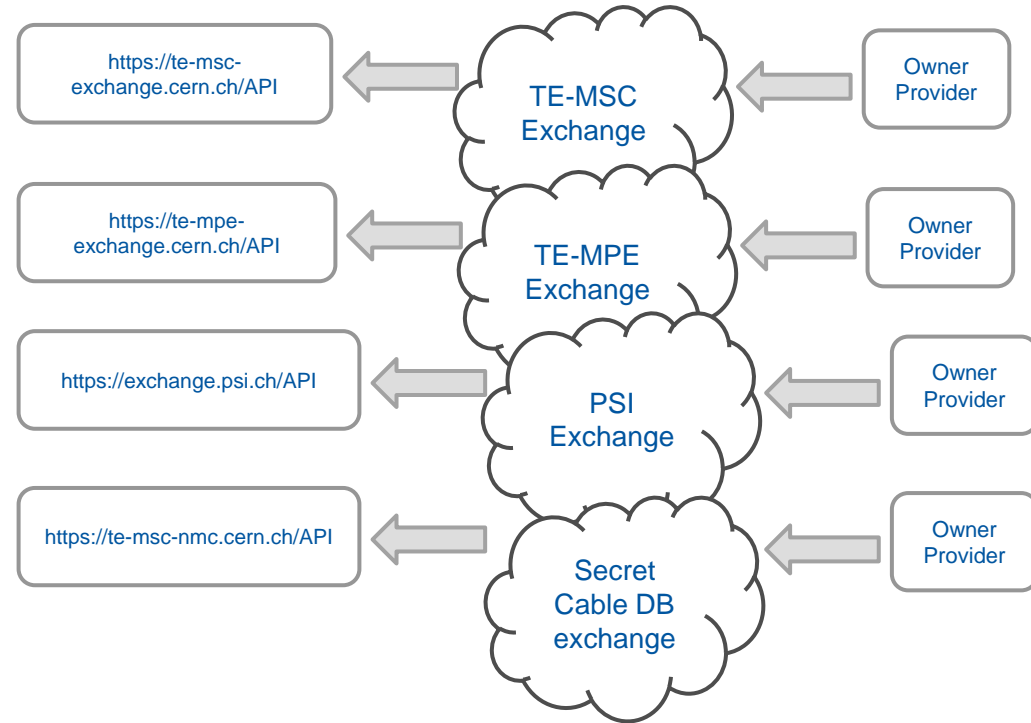
Model Updates
All changes made to the model over time.

- Model created by **Stephan Russenschuck** 22 hours, 21 minutes ago
- Jens Kaeske** added tags: **Dipole** 6h ago
- Stephan Russenschuck** Commented 1h ago

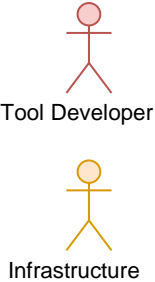
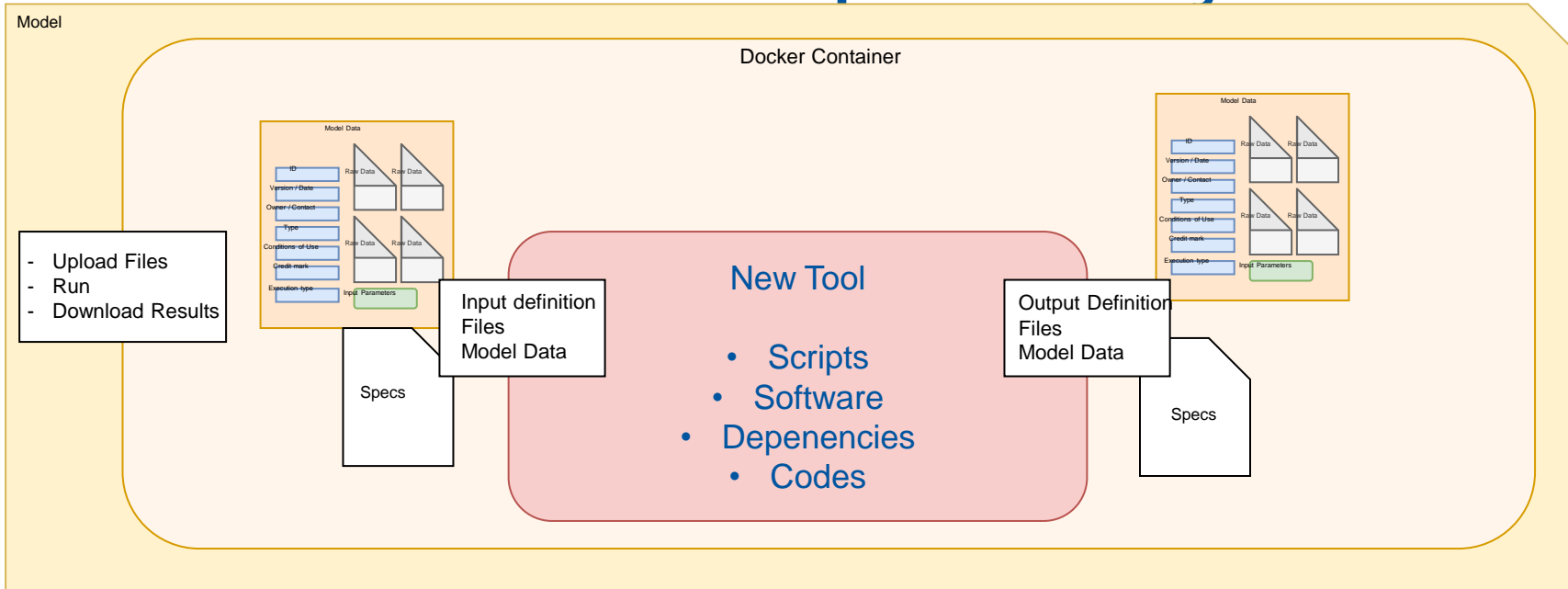
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Tincidunt nunc ipsum tempor purus vitae id. Morbi in vestibulum nec varius. Et diam cursus quis sed purus nam. Selenisque amet elit non sit ut tincidunt condimentum. Nisi ultrices eu venenatis diam.

MMBSE - Boundary crossing

- Separate by responsibility,
 - and End points
- Host separately
 - Access restrictions
 - Data restrictions
 - Computing Resources



MMBSE - Developer Story



Summary

- New Options to extend ROXIE
- Methodology to combine design models
- Environment for Sustainable Magnet development