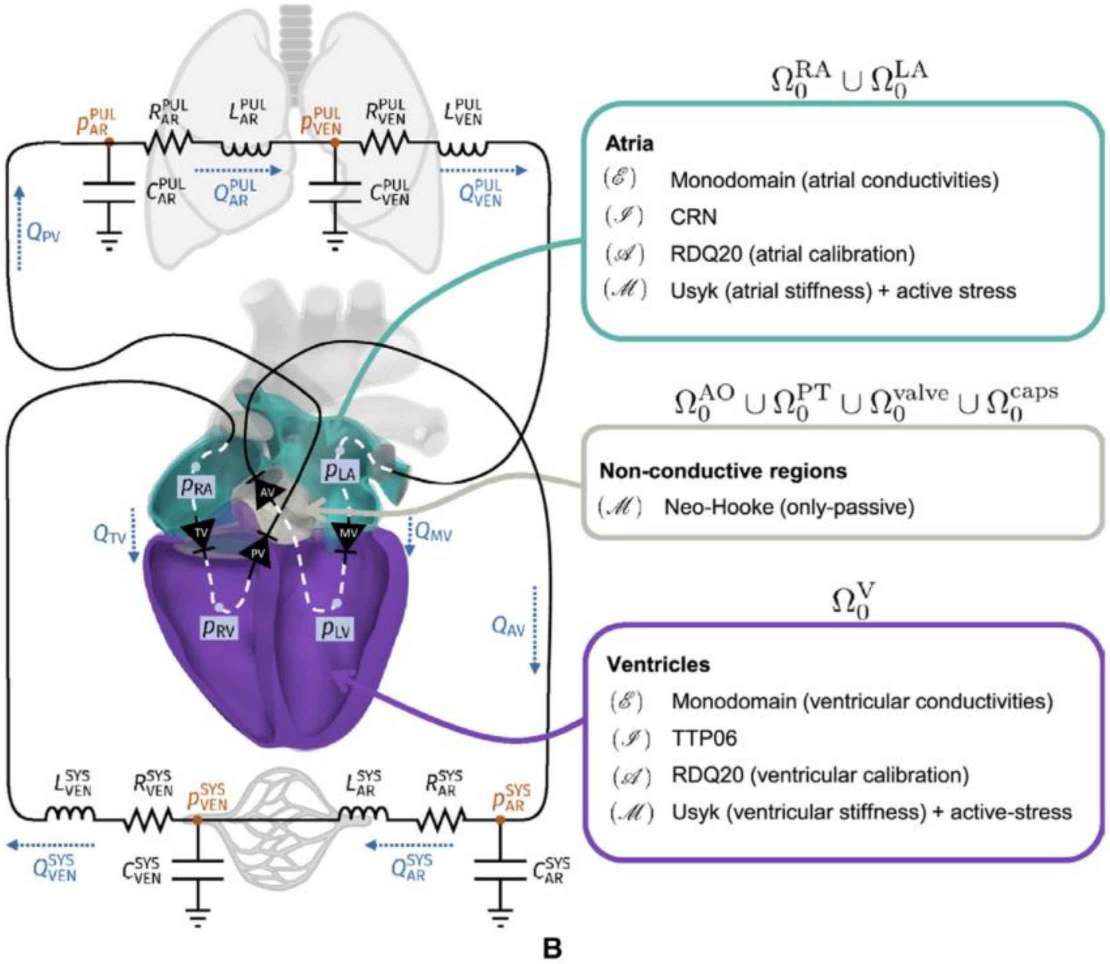
CARDIAC MESH GENERATOR

Vinay Jani

BACKGROUND – DIGITAL TWINS

- Digital Twins are virtual representations of a patient with real-time updates of phenotypic data for precision medicine.
- These involve the generation of 3D models. However, generation of said models has many problems, including:
  - Lack of standardization
  - No co-registration of multiple views
  - Imaging Data (CT, MRI) are 2D.



Coorey et. al. NPJ Digital Med. 2022

BACKGROUND – MESH GENERATION AND FINITE ELEMENT ANALYSIS

Image Co-Registration and Universal Coordinate Transformation

$$\begin{bmatrix} P_x \\ P_y \\ P_z \\ 1 \end{bmatrix} = \begin{bmatrix} X_x\Delta_i & Y_x\Delta_j & 0 & S_x \\ X_y\Delta_i & Y_y\Delta_j & 0 & S_y \\ X_z\Delta_i & Y_z\Delta_j & 0 & S_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} i \\ j \\ 0 \\ 1 \end{bmatrix} = M \begin{bmatrix} i \\ j \\ 0 \\ 1 \end{bmatrix}$$

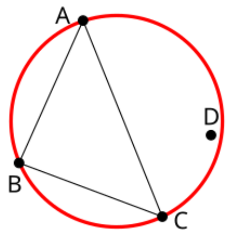
absolute coordinates

image indices

pixel spacing; direction cosines

Mesh Generation and Delaunay Triangulation

$$\begin{vmatrix} A_x & A_y & A_x^2 + A_y^2 & 1 \\ B_x & B_y & B_x^2 + B_y^2 & 1 \\ C_x & C_y & C_x^2 + C_y^2 & 1 \\ D_x & D_y & D_x^2 + D_y^2 & 1 \end{vmatrix}$$



$$= \begin{vmatrix} A_x - D_x & A_y - D_y & (A_x - D_x)^2 + (A_y - D_y)^2 \\ B_x - D_x & B_y - D_y & (B_x - D_x)^2 + (B_y - D_y)^2 \\ C_x - D_x & C_y - D_y & (C_x - D_x)^2 + (C_y - D_y)^2 \end{vmatrix} > 0$$

Parameter Optimization: Elliptic PDEs

$$\alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} = -I^2(Px_\xi + Qx_\eta)$$

$$\alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} = -I^2(Py_\xi + Qy_\eta)$$

$$\alpha = x_\eta^2 + y_\eta^2$$

$$\beta = x_\eta x_\xi + y_\xi y_\eta$$

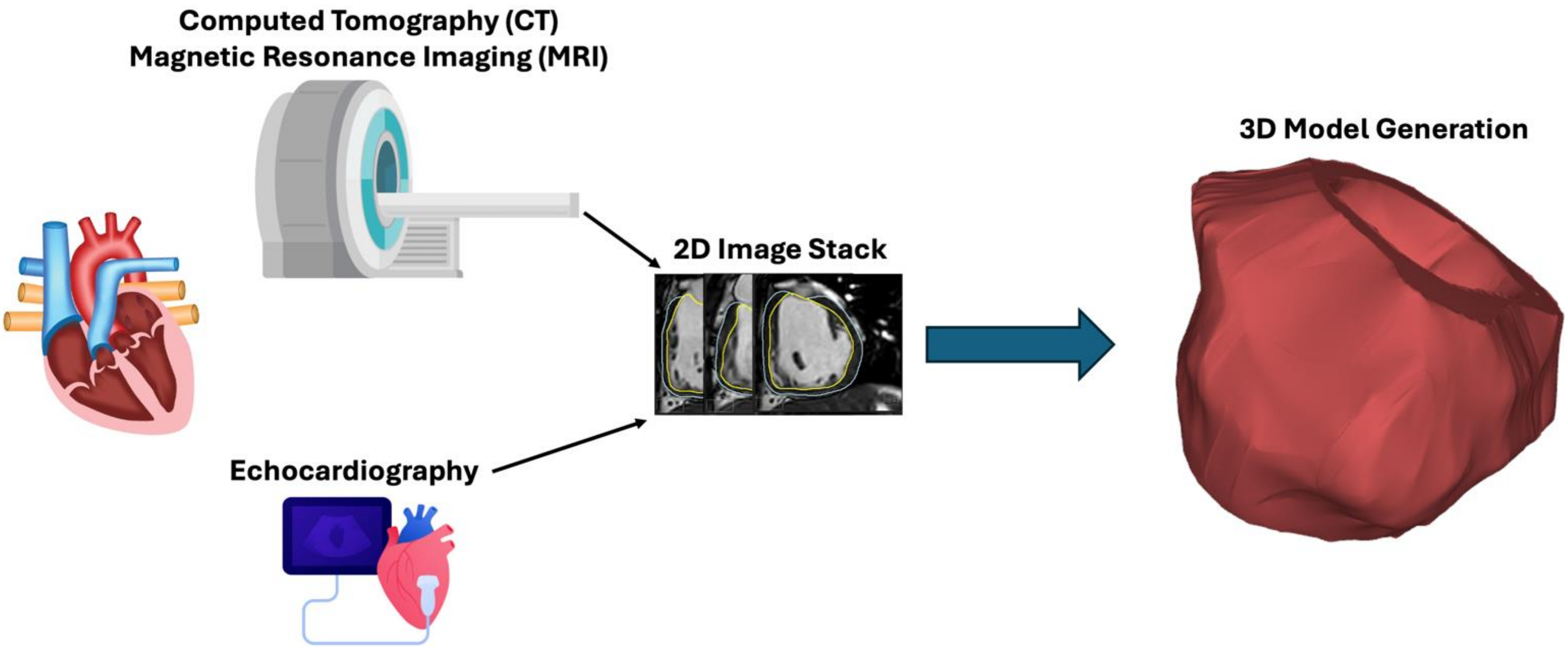
$$\gamma = x_\xi^2 + y_\xi^2$$

$$I = \frac{\delta(x,y)}{\delta(\xi,\eta)} = y_\eta x_\xi - y_\xi x_\eta$$

PROBLEM STATEMENT

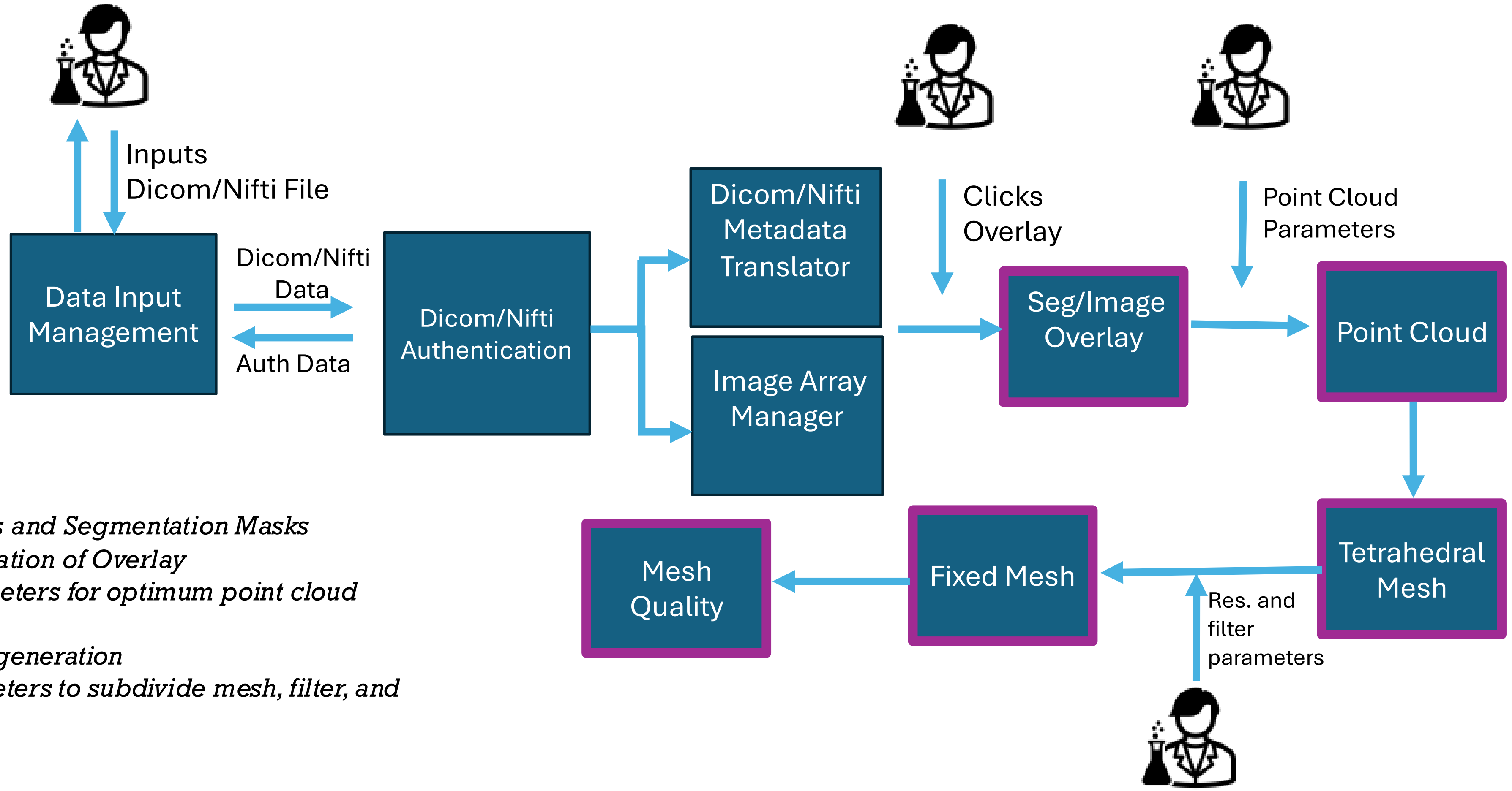
- A means to visualize and generate 3D models from cardiac MRI data , which can colocalize views and is easily accessible does not exist

PROPOSED PIPELINE FOR MODEL GENERATION



USE CASES

User Profile: Non-programmer user who uses patient specific cardiac studies and simulations



1. *User Inputs Dicoms and Segmentation Masks*
2. *User clicks visualization of Overlay*
3. *User selects parameters for optimum point cloud generation*
4. *User selects mesh generation*
5. *User inputs parameters to subdivide mesh, filter, and fix mesh*

METHODOLOGY

Overlay and Image Processing

DICOMS
AND
SEGMENTATIONS

→

defImportDicomSeries
Import and Localize pixels
defGetMasks- Import and
Localize Mask Pixels

→

defgetMaskOverlay
Overlay segmentations with
image stack

Point Cloud Generation

Processed Pixel Matricies
from above

→

defGeneratePointCloud
Generates, Cleans, and
Vizualizes Point Cloud

→

defSavePointCloud
Saves a ply of point cloud

Tetrahedral Mesh

Point Cloud from above

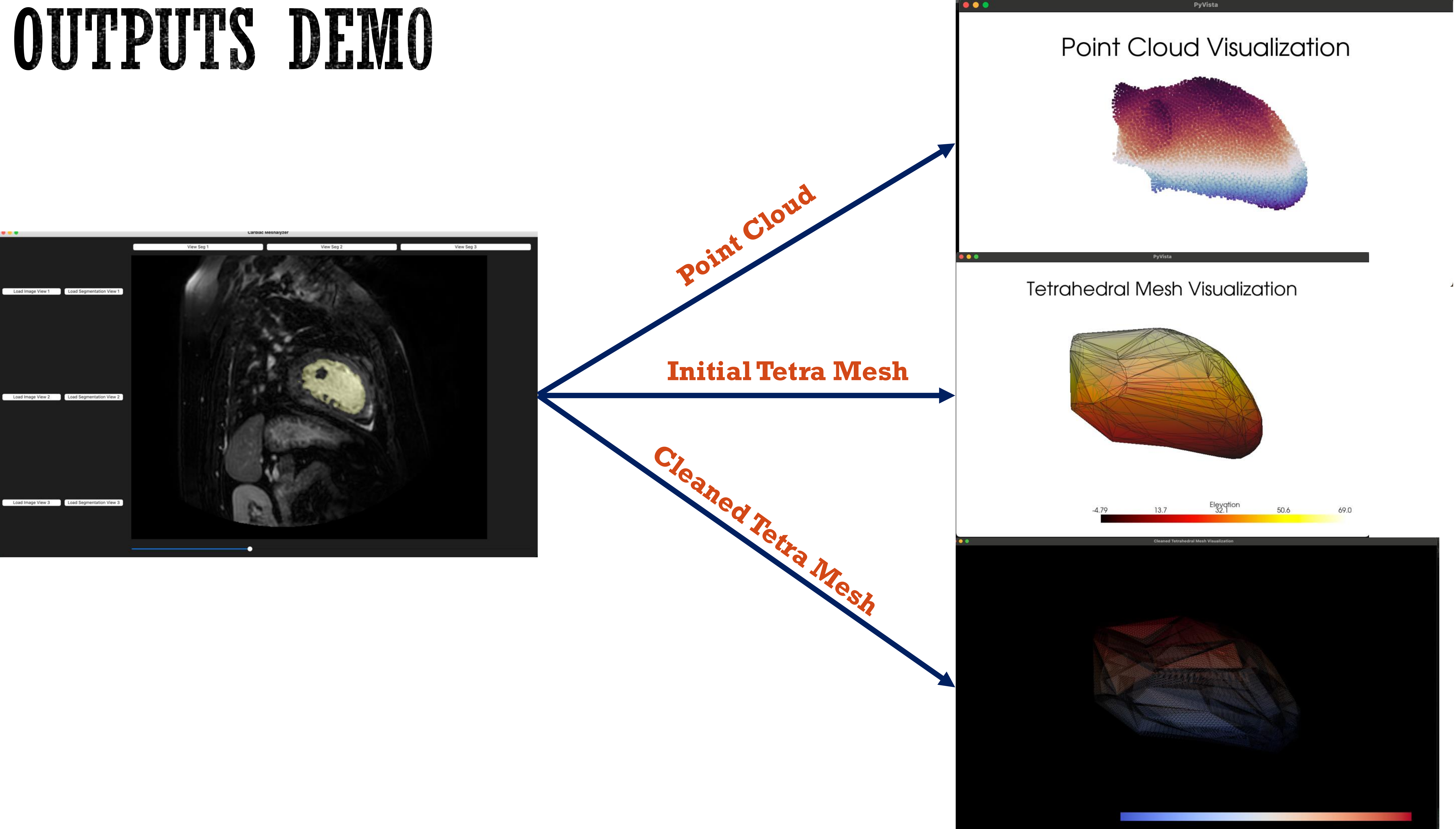
→

defgeneratetetramesh
Delauney Triangulation

→

defclean_tetra_mesh
Poisson Smoothing and
Subdivison
defsavemesh- saves as VTK

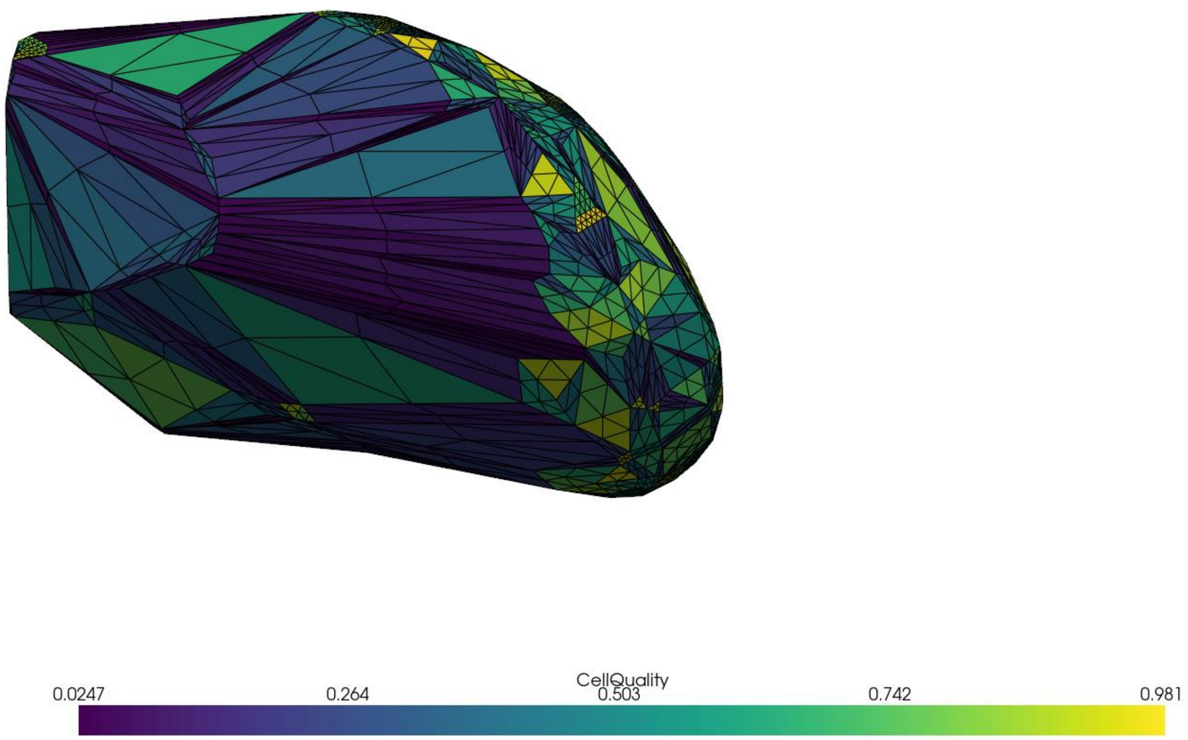
OUTPUTS DEMO



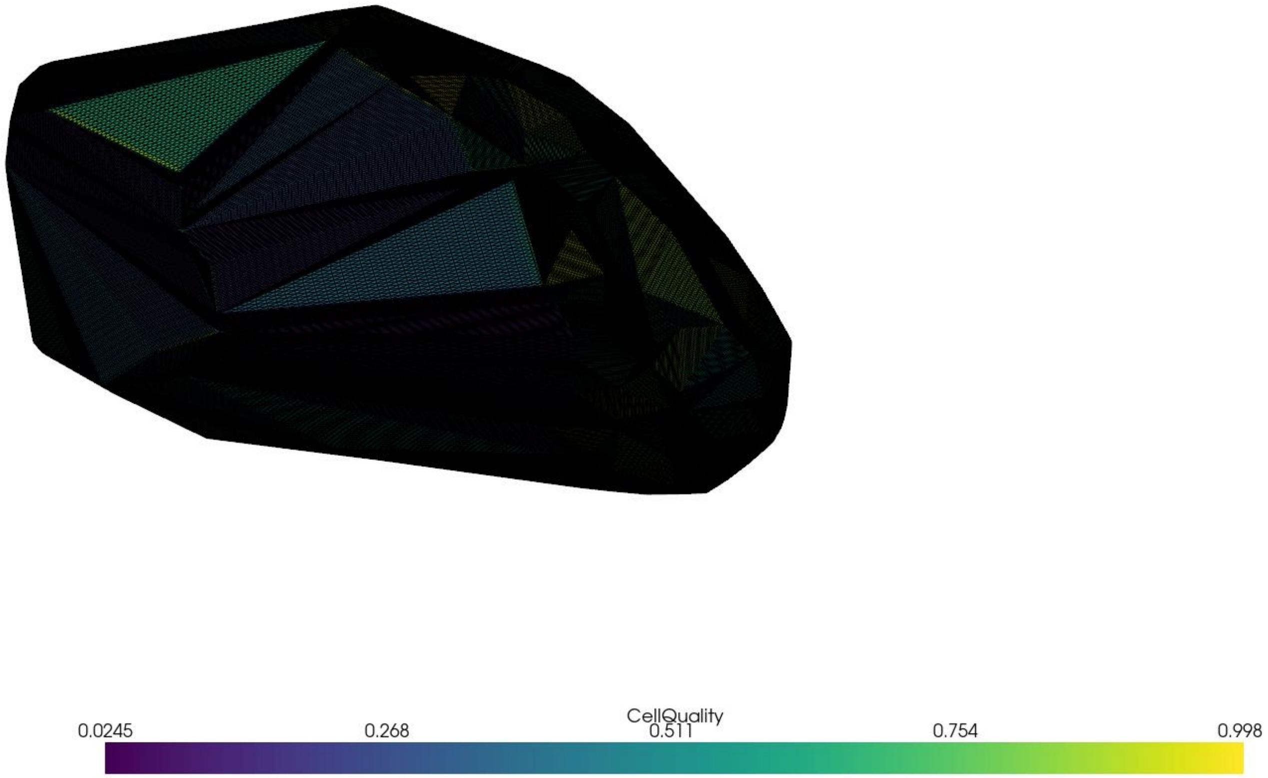
RESULTS

- **Speed – Can make mesh in less than 1 second and perform a 100 interactions Poisson with 5 subdivisions in less than 10 seconds**
- **Filtering and cleaning methods allow for increased resolution**

No Smoothing or Filtering

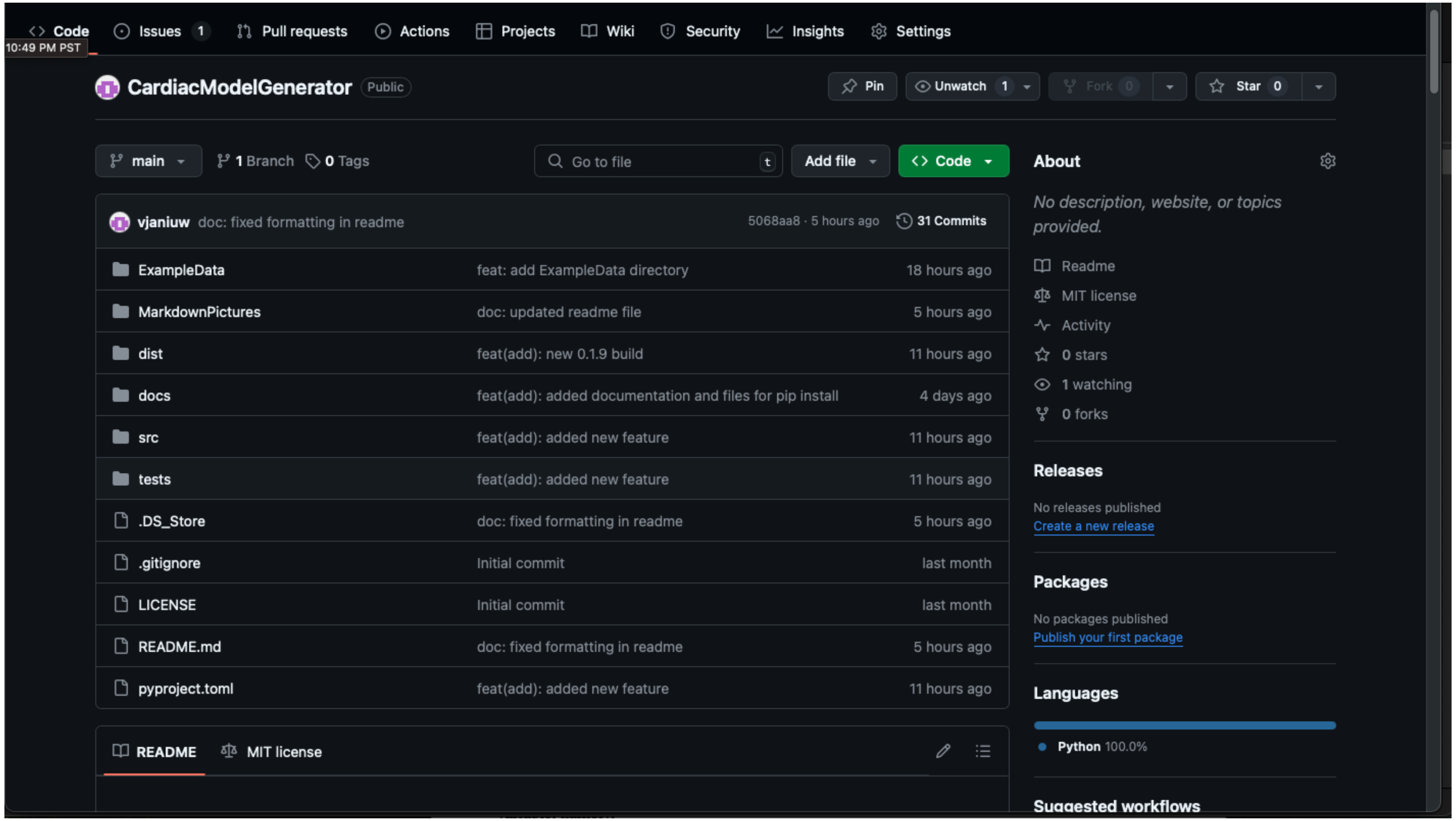


New Smoothing/Subdivision Algorithm



PROJECT STRUCTURE

```
CardiacModelGenerator/
├── ExampleData/
├── MarkdownPictures/
├── dist/
├── docs/
├── src/
├── tests/
├── .gitignore
├── LICENSE
├── README.md
├── pyproject.toml
```



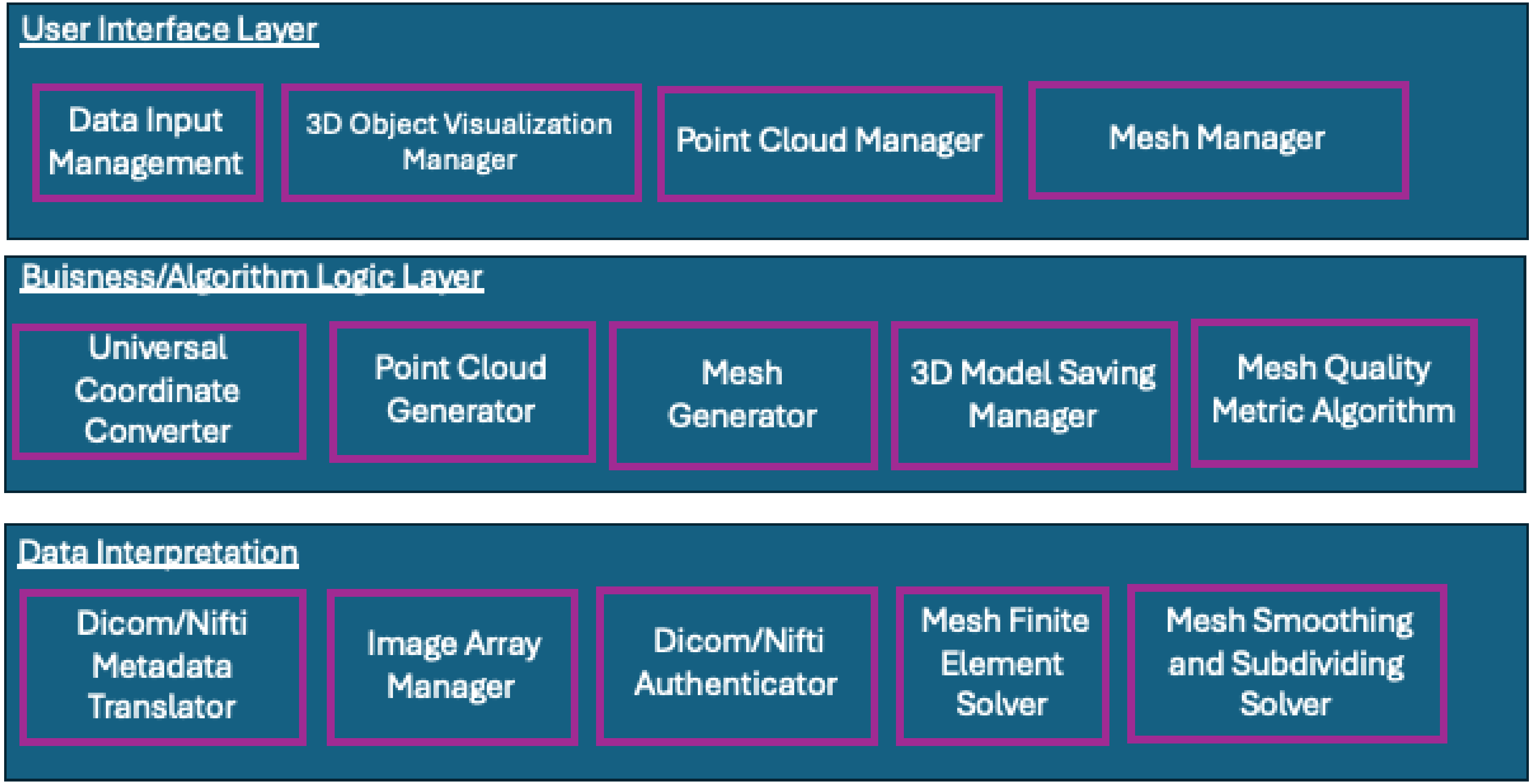
https://github.com/vjaniuw/CardiacModelGenerator

CHALLENGES

1. Object Oriented Programming with WX
2. Meshing Computationally Expensive
3. User Configurable Meshing is non-uniform
4. Co-registration of images in real coordinates
5. Image Segmentation input requires expertise
6. Rigidity of Medical Image Data

APPENDIX

DESIGN INPUTS- LAYERED ARCHITECTURE



DESIGN OUTPUT