

# **1. Block 4.2: CS6514: Software Architecture Design Portfolio**

Fluctuating Finite Element Analysis (FFEA) Case Study

Art Ó Liathain

September 2025

## 2. Table of Contents

### Contents

1. Block 4.2: CS6514: Software Architecture Design Portfolio .....	1
2. Table of Contents .....	2
3. Tech Stack .....	3
4. Utility Tree .....	5
5. Use Case Diagram .....	5
6. 4+1 Diagram .....	6
6.1. Logical View .....	6
6.2. Process view .....	6
6.3. Physical View .....	6
6.4. Development View .....	6
7. Codescene .....	7
8. Understand .....	8
9. Sonarqube .....	10
10. C4 .....	12

### List of Figures

Figure 1 Use Case Diagram .....	5
Figure 2 Dependency Coupling Graph .....	7
Figure 3 Technical Debt Codescene .....	8
Figure 4 Understand High Level Info .....	8
Figure 5 Understand complexity ratings .....	9
Figure 6 Class Diagram Inheritance .....	9
Figure 7 Class Diagram flat .....	9
Figure 8 .....	10
Figure 9 .....	10
Figure 10 .....	11
Figure 11 .....	11
Figure 12 Context Diagram .....	12
Figure 13 Component Diagram .....	12
Figure 14 Container Diagram .....	12

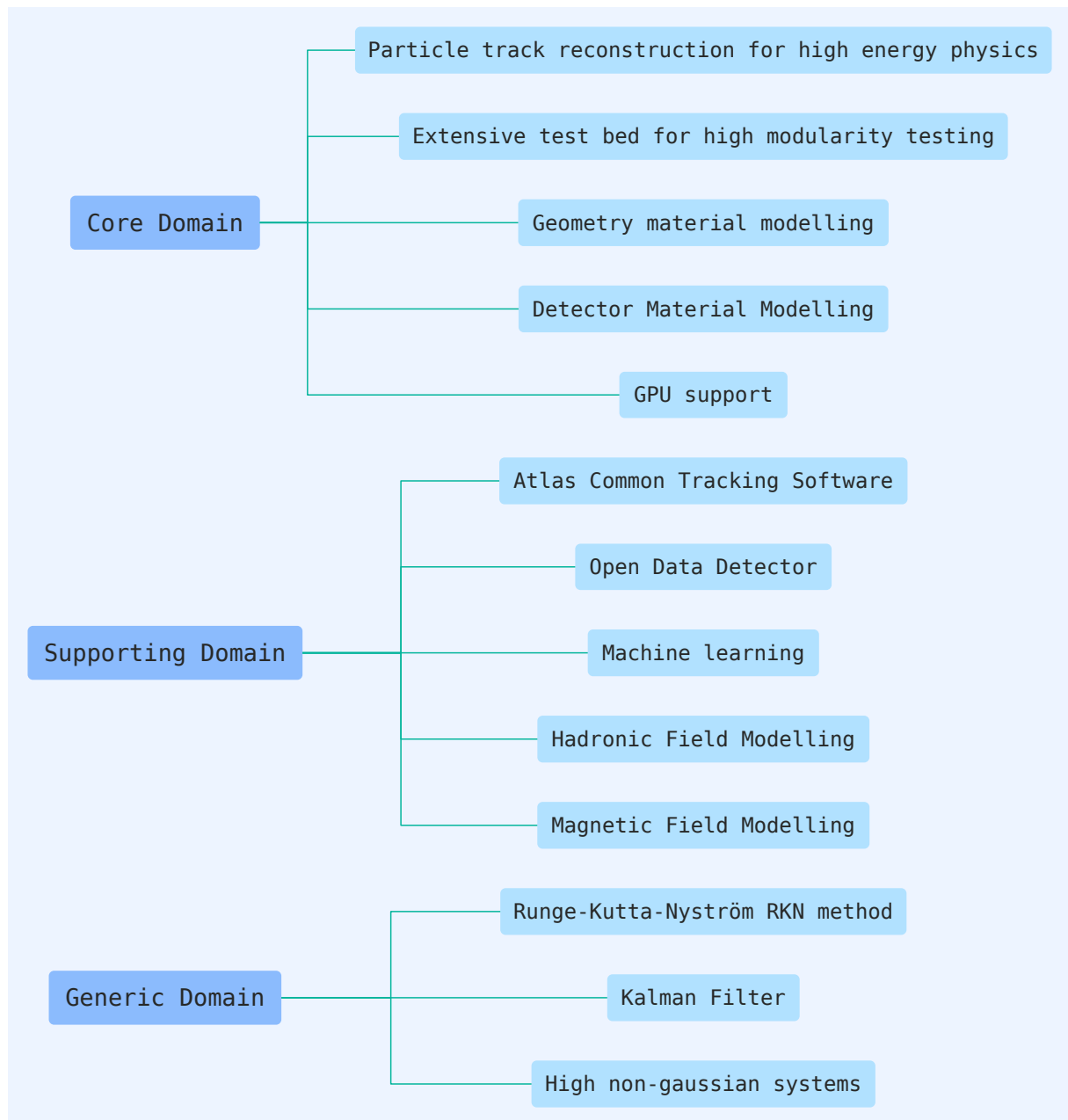
### **3. Tech Stack**

Main Language: C++

Heavily used external libraries: Eigen, boost, cmake

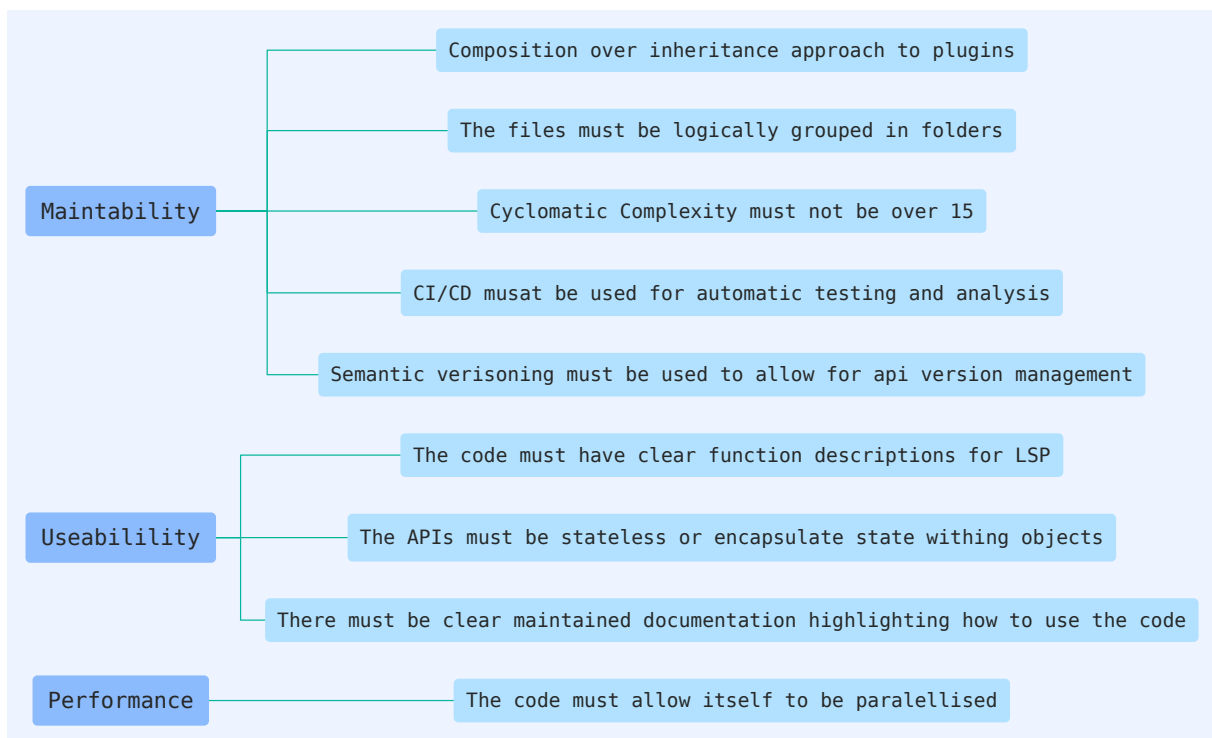
Used for example scripts: Python bindings using pybind11

Containerisation: Docker



Listing 1: Domain Model

## 4. Utility Tree



Listing 2: Utility Tree

## 5. Use Case Diagram

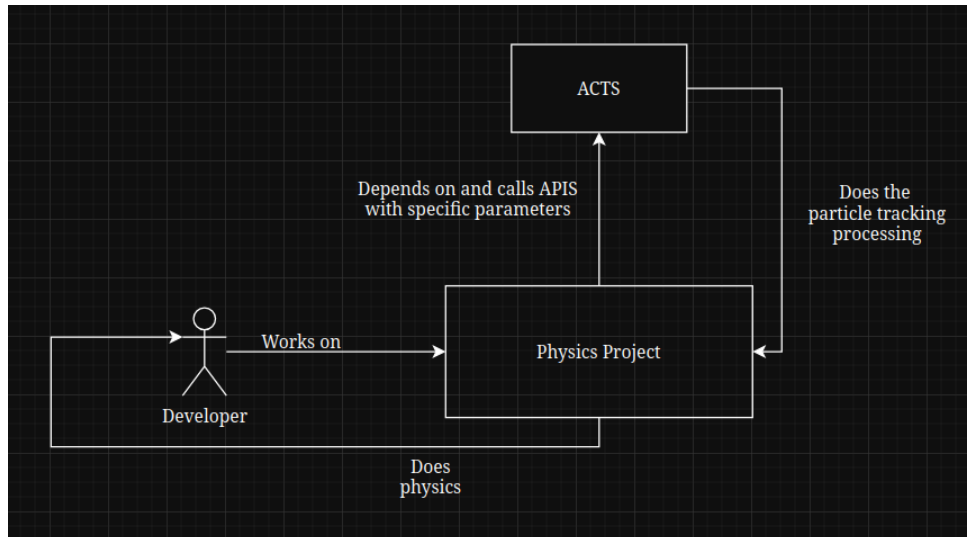
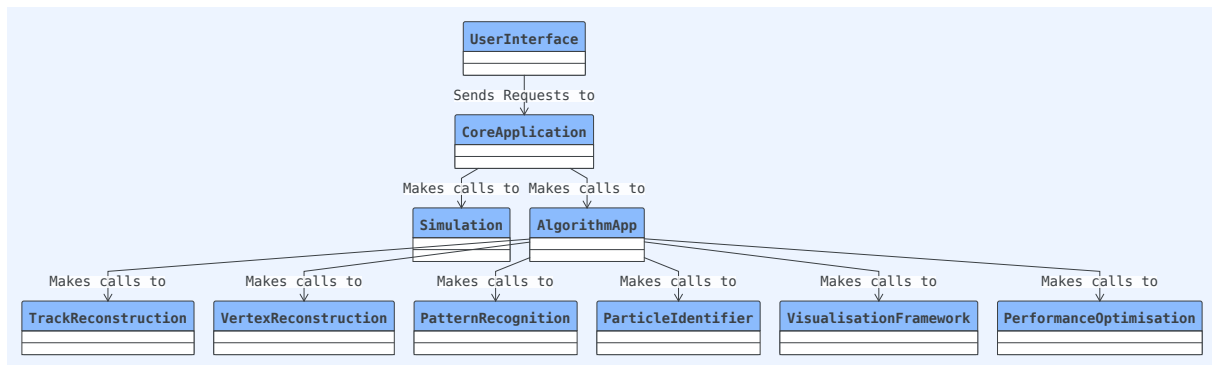


Figure 1: Use Case Diagram

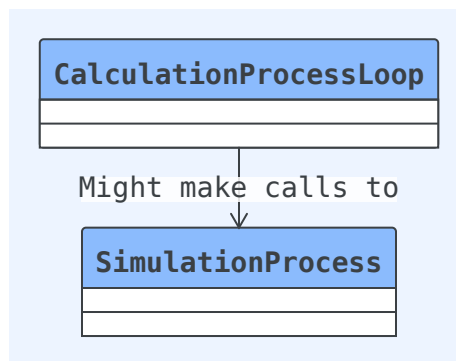
## 6. 4+1 Diagram

### 6.1. Logical View



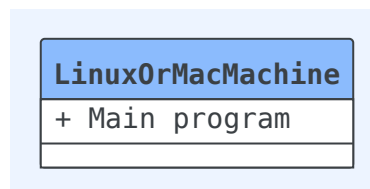
Listing 3:

### 6.2. Process view



Listing 4: Process view

### 6.3. Physical View



### 6.4. Development View

- There are two layers of folder organisation in the codebase. The first is include/ and src/ following that there are logical groupings such as Geometry/ but there are some subfolders with a high file count befitting another layer of depth which is not in place. Geometry/ has over 30 files flat
- There is a consistent commit structure and rule
- Integration tests are loosely handled in

## 7. Codescene



Figure 2: Dependency Coupling Graph

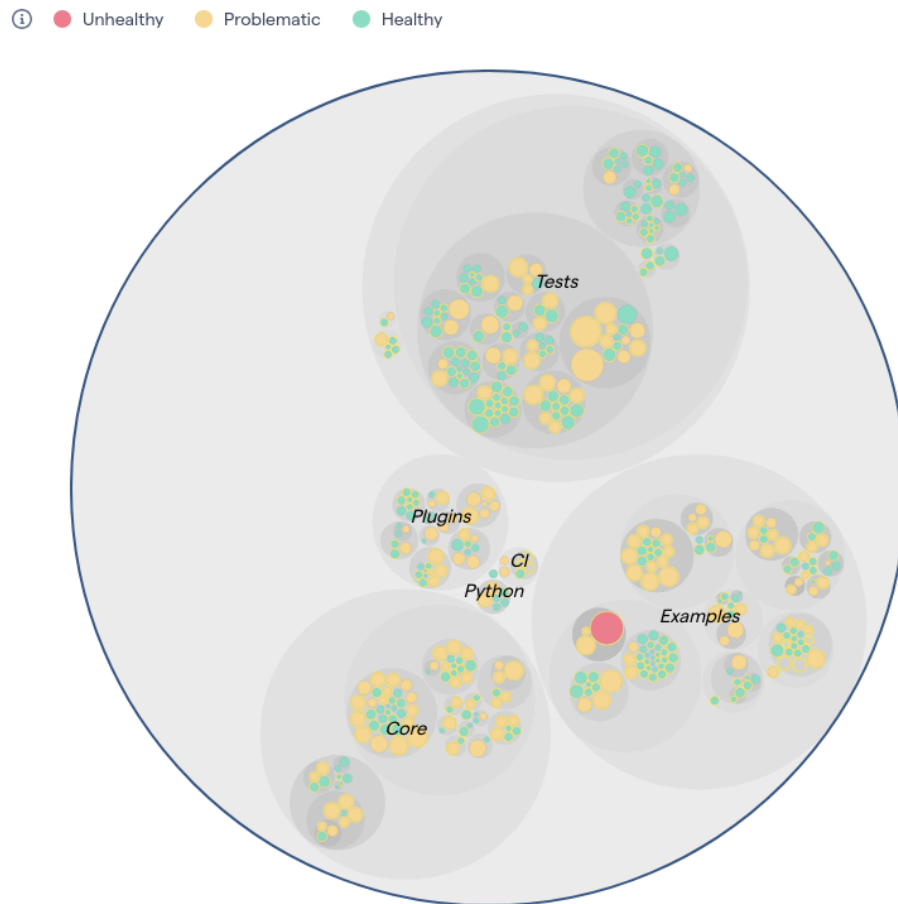


Figure 3: Technical Debt Codescene

## 8. Understand

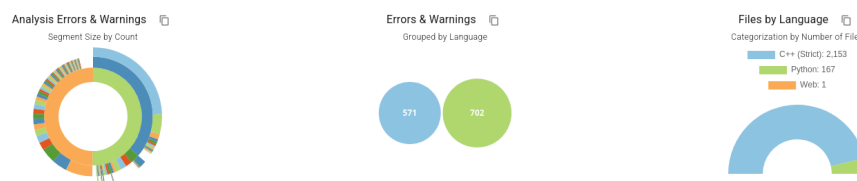


Figure 4: Understand High Level Info





## 9. Sonarqube

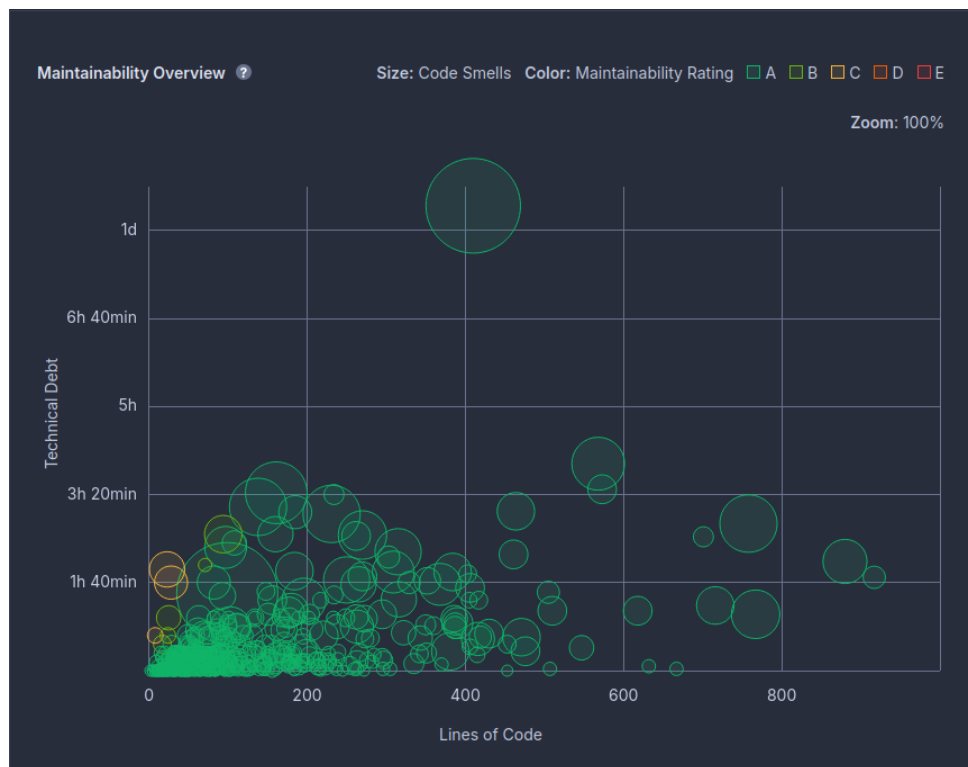


Figure 8:

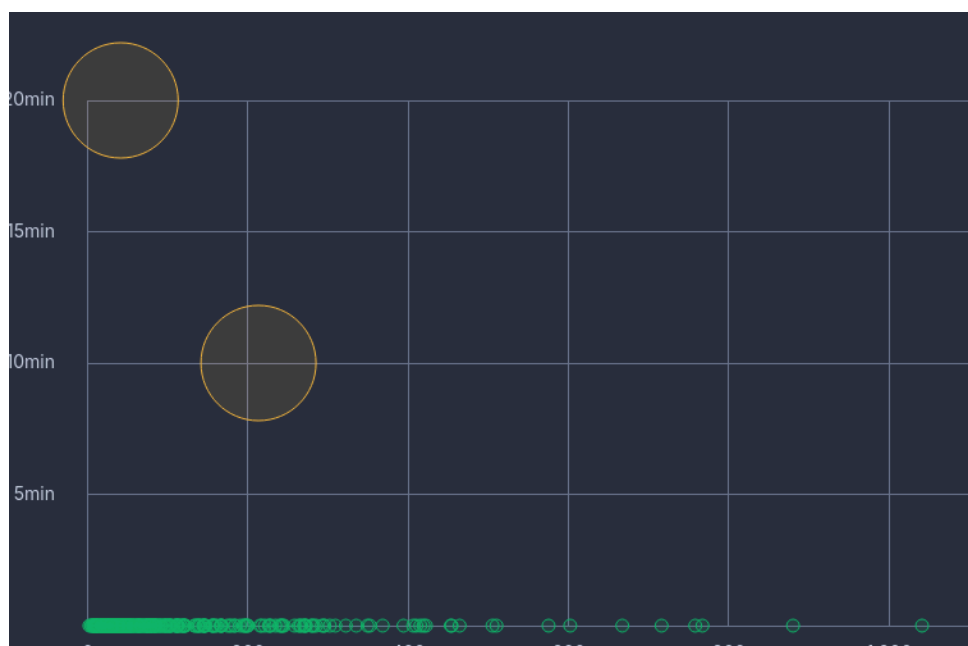


Figure 9:

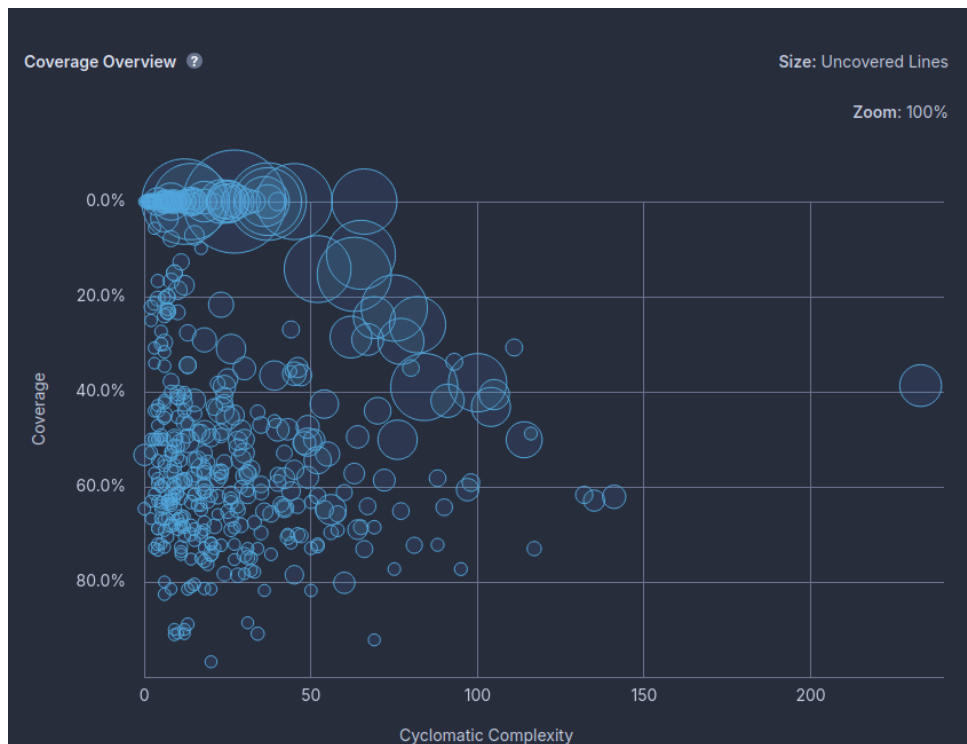



Figure 10:

acts > Core > src > Geometry  View as List 60 files

**Cyclomatic Complexity 2,133** New code: since v44.0.0















 Core/src/Geometry/TrackingVolume.cpp	135
 Core/src/Geometry/CylinderVolumeStack.cpp	132
 Core/src/Geometry/GridPortalLink.cpp	111
 Core/src/Geometry/CylinderVolumeHelper.cpp	100
 Core/src/Geometry/CuboidVolumeStack.cpp	98
 Core/src/Geometry/CylinderPortalShell.cpp	97
 Core/src/Geometry/CylinderVolumeBuilder.cpp	82
 Core/src/Geometry/GridPortalLinkMerging.cpp	80
 Core/src/Geometry/Portal.cpp	77
 Core/src/Geometry/detail/MaterialDesignator.hpp	70
 Core/src/Geometry/CuboidPortalShell.cpp	67
 Core/src/Geometry/CompositePortalLink.cpp	52
 Core/src/Geometry/ConeVolumeBounds.cpp	52
 Core/src/Geometry/Layer.cpp	50

Figure 11:

## 10. C4

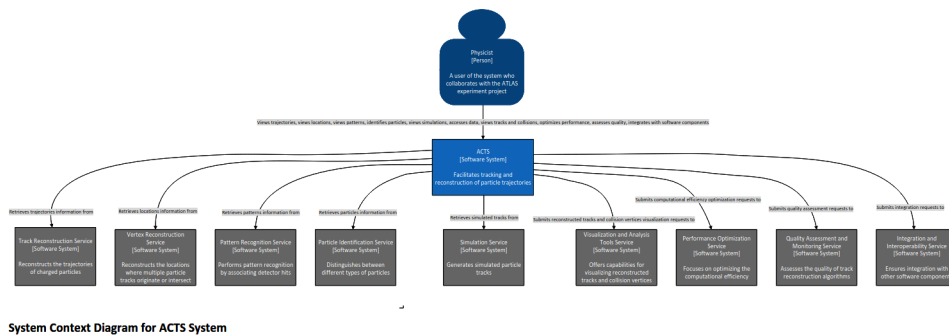


Figure 12: Context Diagram

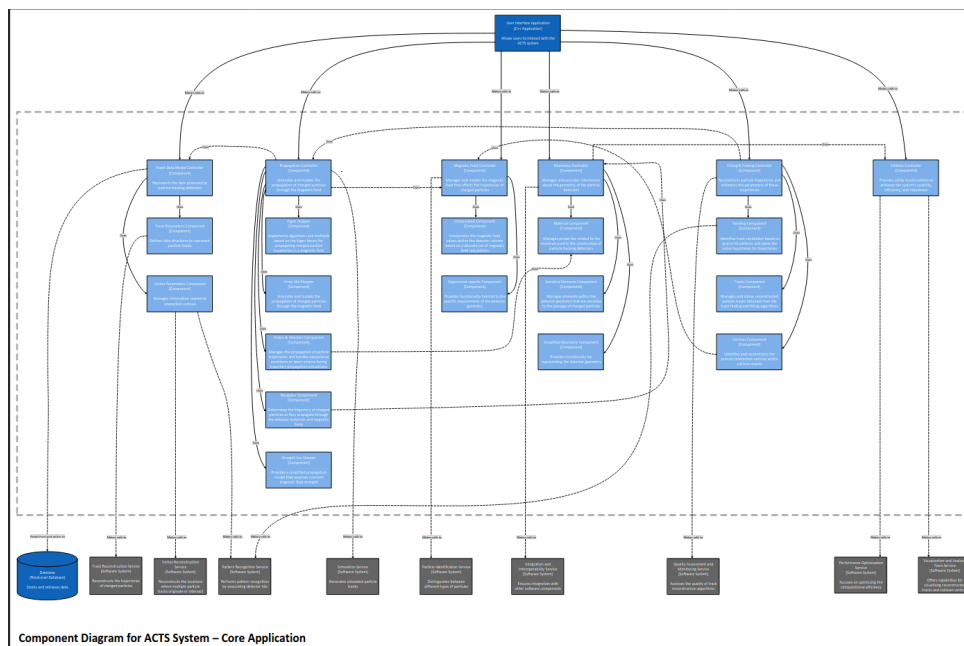


Figure 13: Component Diagram

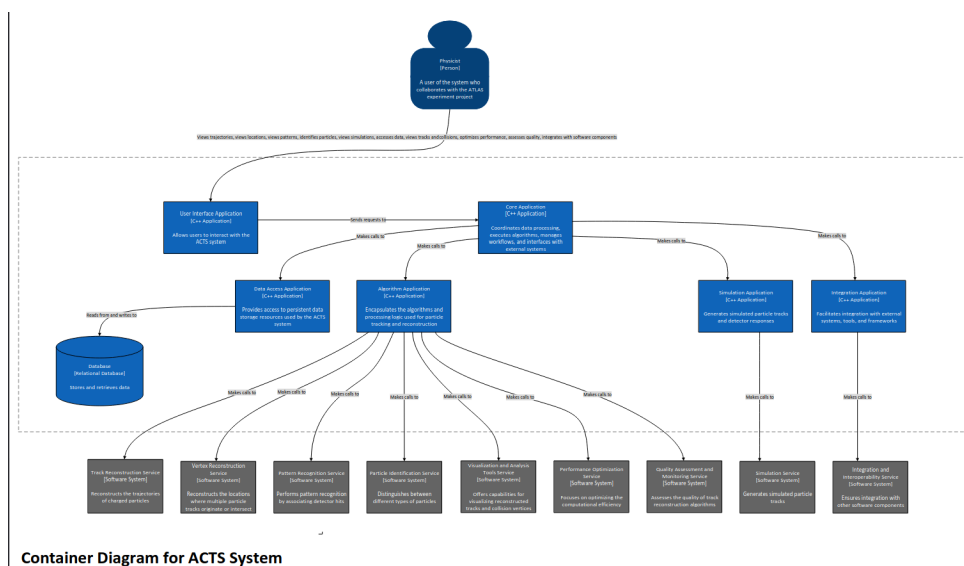


Figure 14: Container Diagram