Designing modular software for template fitting photo-z estimation

Presentation by Nikolaos Apostolakos
(Nikolaos.Apostolakos@unige.ch)

What is Phosphoros

What is Phosphoros

- Template fitting photo-z tool for Euclid

What is Phosphoros

- Template fitting photo-z tool for Euclid
- C++11 project with Python help tools

What is Phosphoros

- Template fitting photo-z tool for Euclid
- C++11 project with Python help tools
- Use in Euclid production pipeline
- Use for scientific algorithm exploration

What is Phosphoros

- Template fitting photo-z tool for Euclid
- C++11 project with Python help tools
- Use in Euclid production pipeline
- Use for scientific algorithm exploration
- It is designed to be:
  - Modular
  - Flexible
  - Robust
  - Reusable

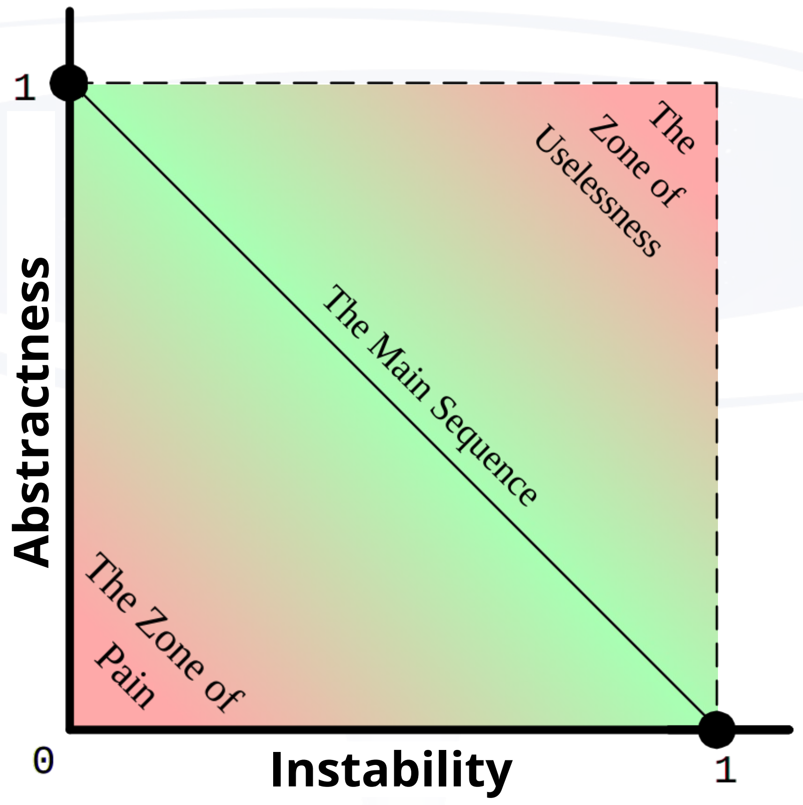
What is Phosphoros

- Template fitting photo-z tool for Euclid
- C++11 project with Python help tools
- Use in Euclid production pipeline
- Use for scientific algorithm exploration
- It is designed to be:
  - Modular
  - Flexible
  - Robust
  - Reusable
- Follows the SOLID design principles

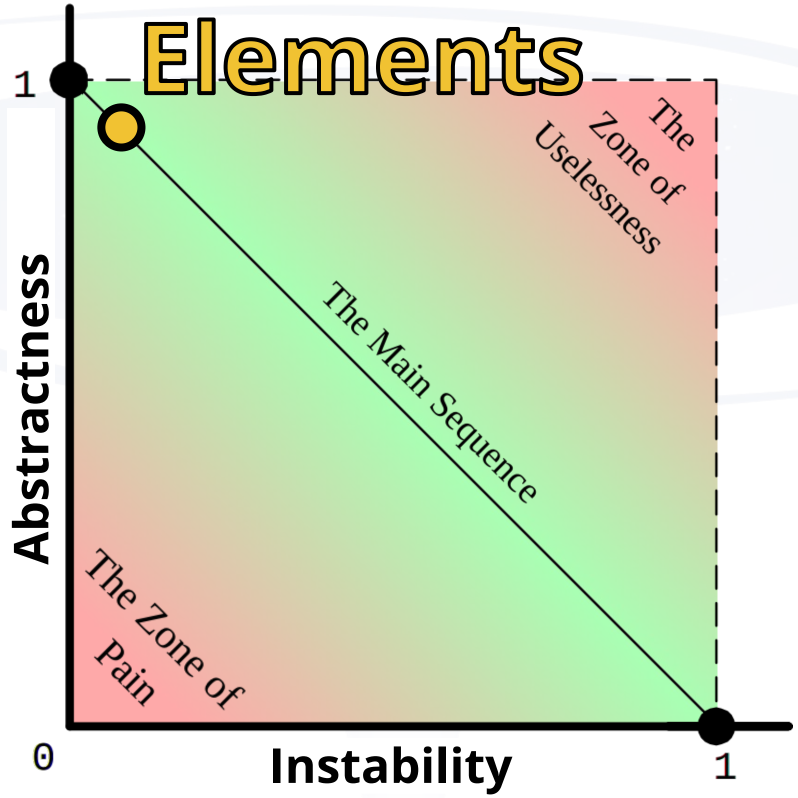
What is Phosphoros

- Template fitting photo-z tool for Euclid
- C++11 project with Python help tools
- Use in Euclid production pipeline
- Use for scientific algorithm exploration
- It is designed to be:
    - Modular
    - Flexible
    - Robust
    - Reusable
- Follows the SOLID design principles
- Can be used as a generic photo-z tool

Project Organization

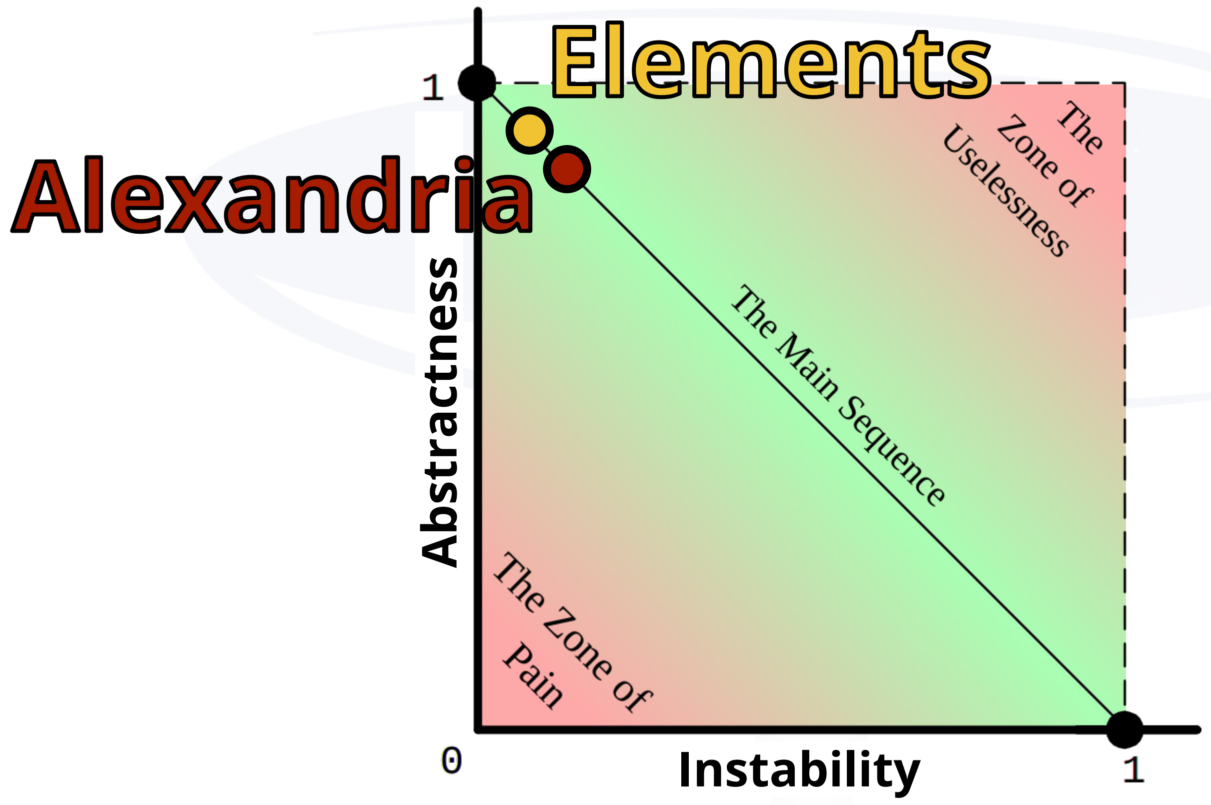


Project Organization



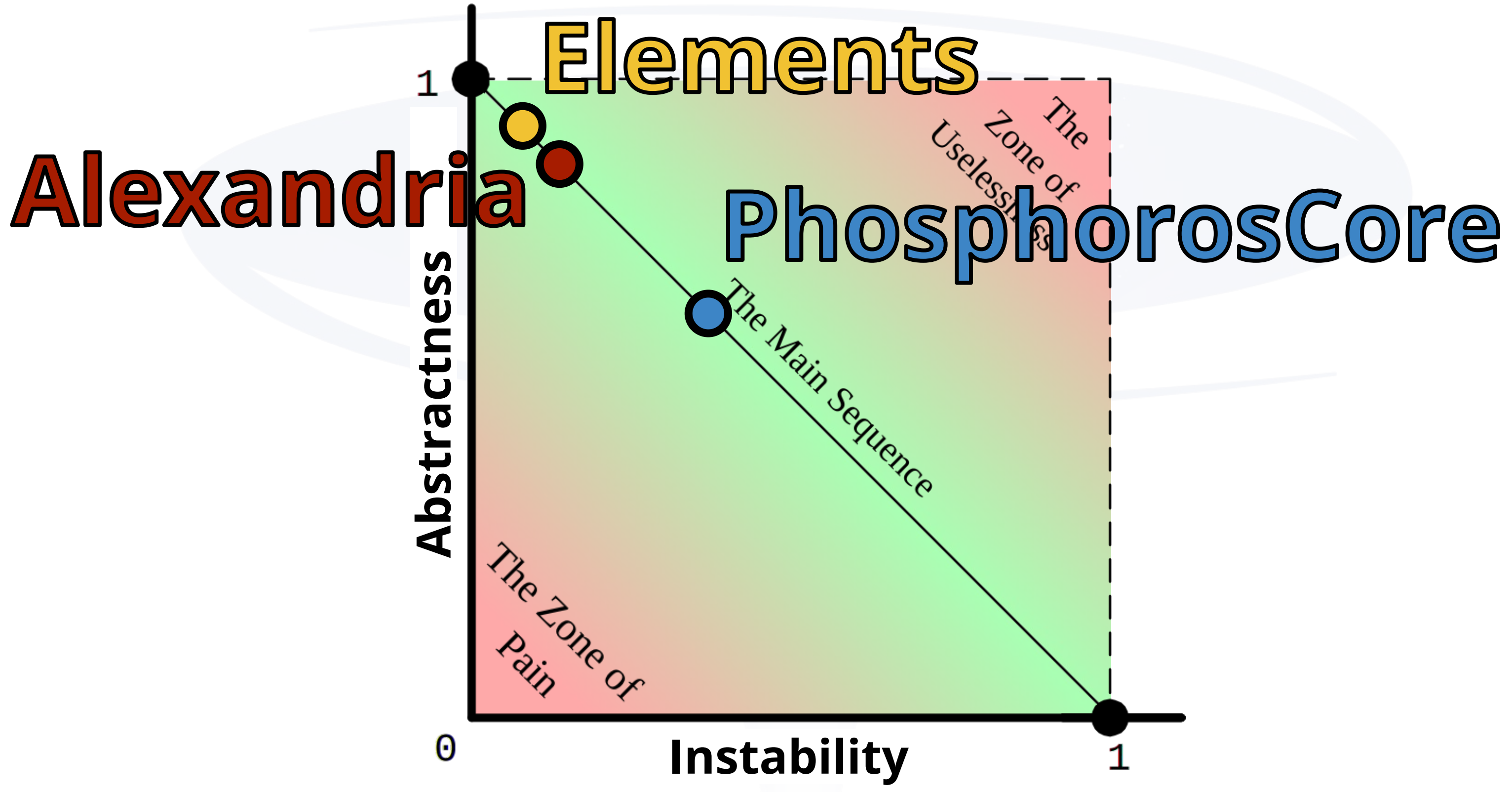
Elements: C++/Python Build and Application Framework for the Euclid mission

Project Organization



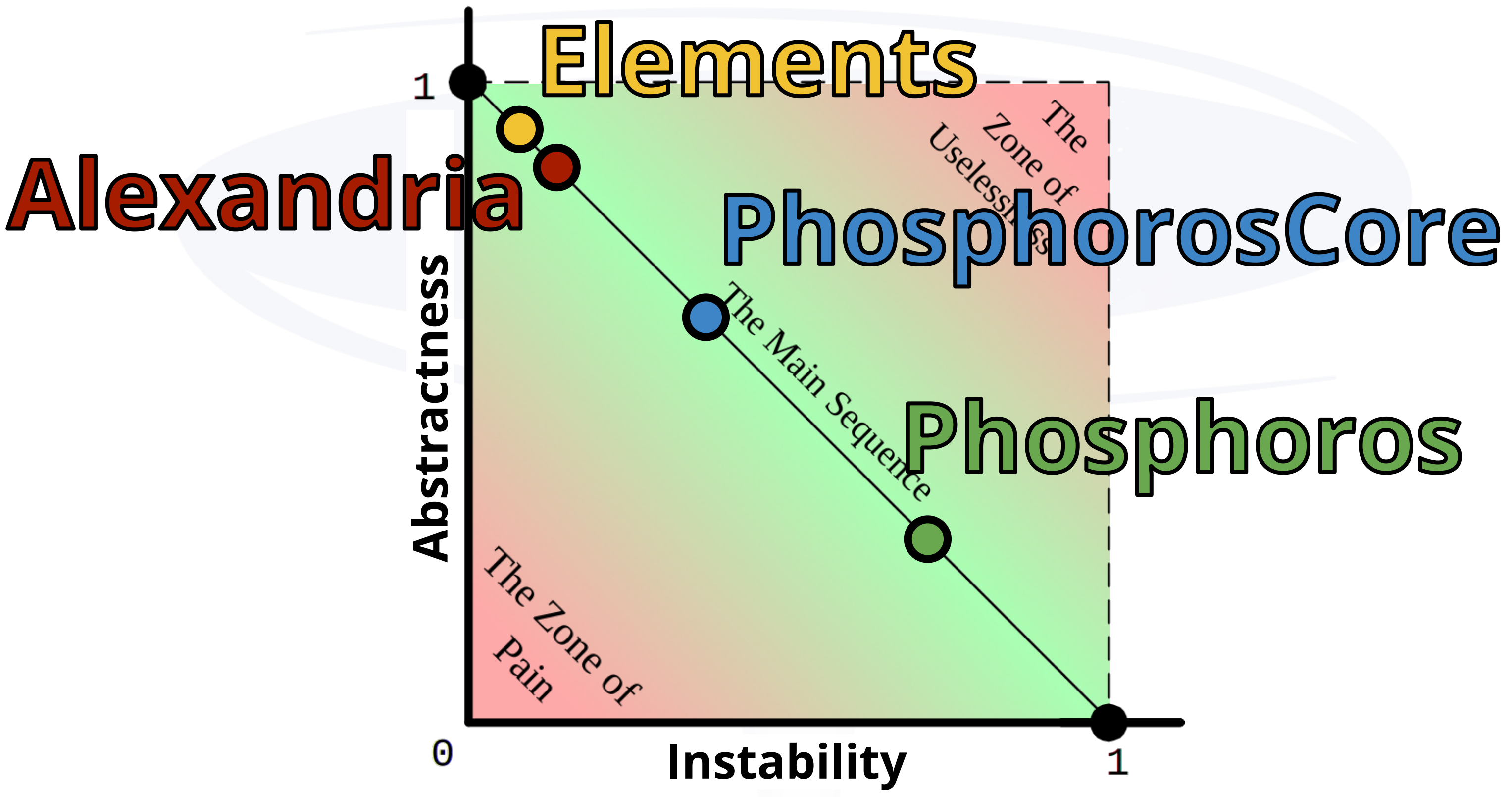
Alexandria: SDC-CH core library, generic modules to be reused with other projects

Project Organization



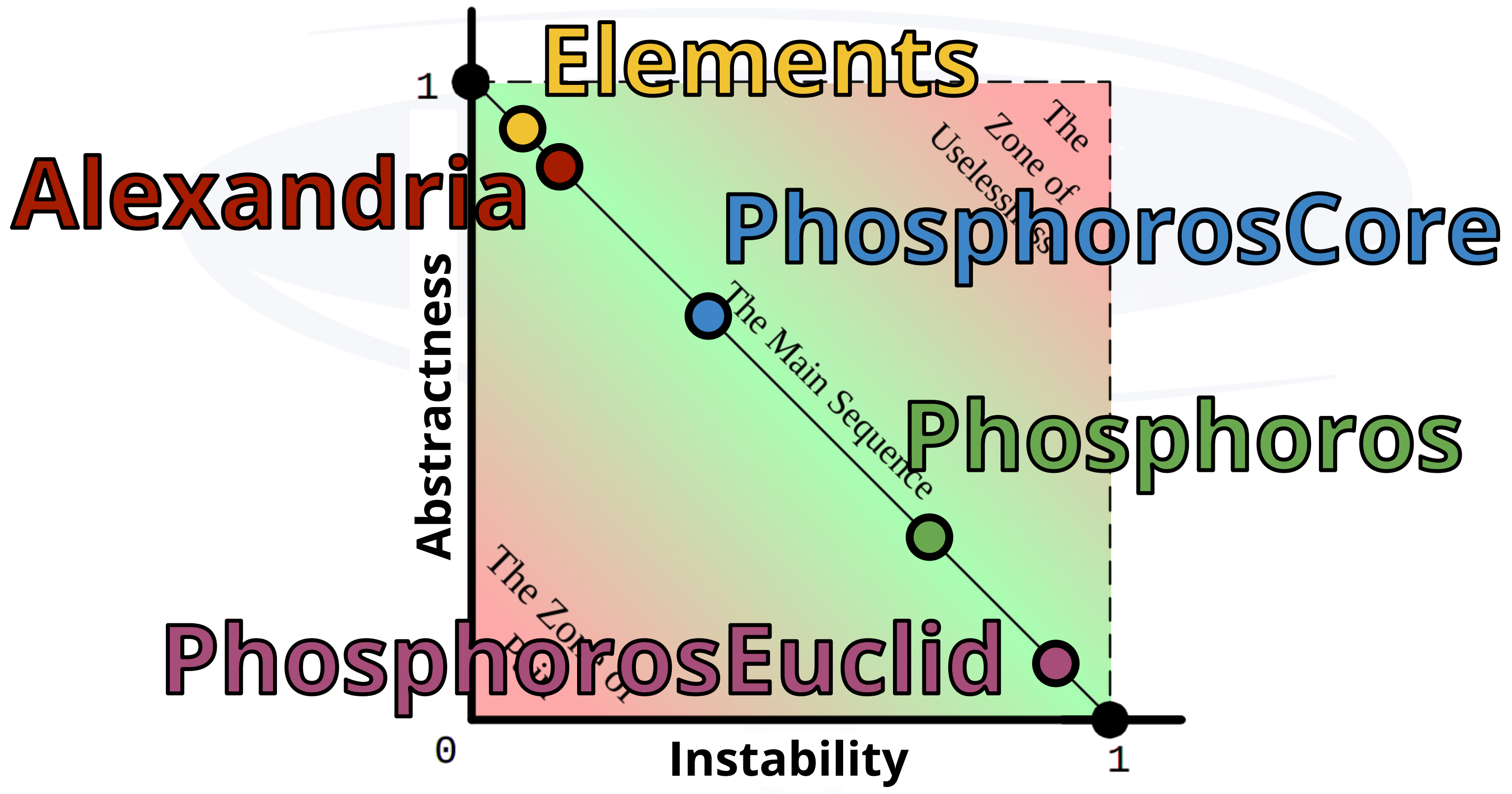
PhosphorosCore: Core template fitting code (including basic executables)

Project Organization



Phosphoros: PHZ post-processing, GUI, utility tools (constitutes Phosphoros product)

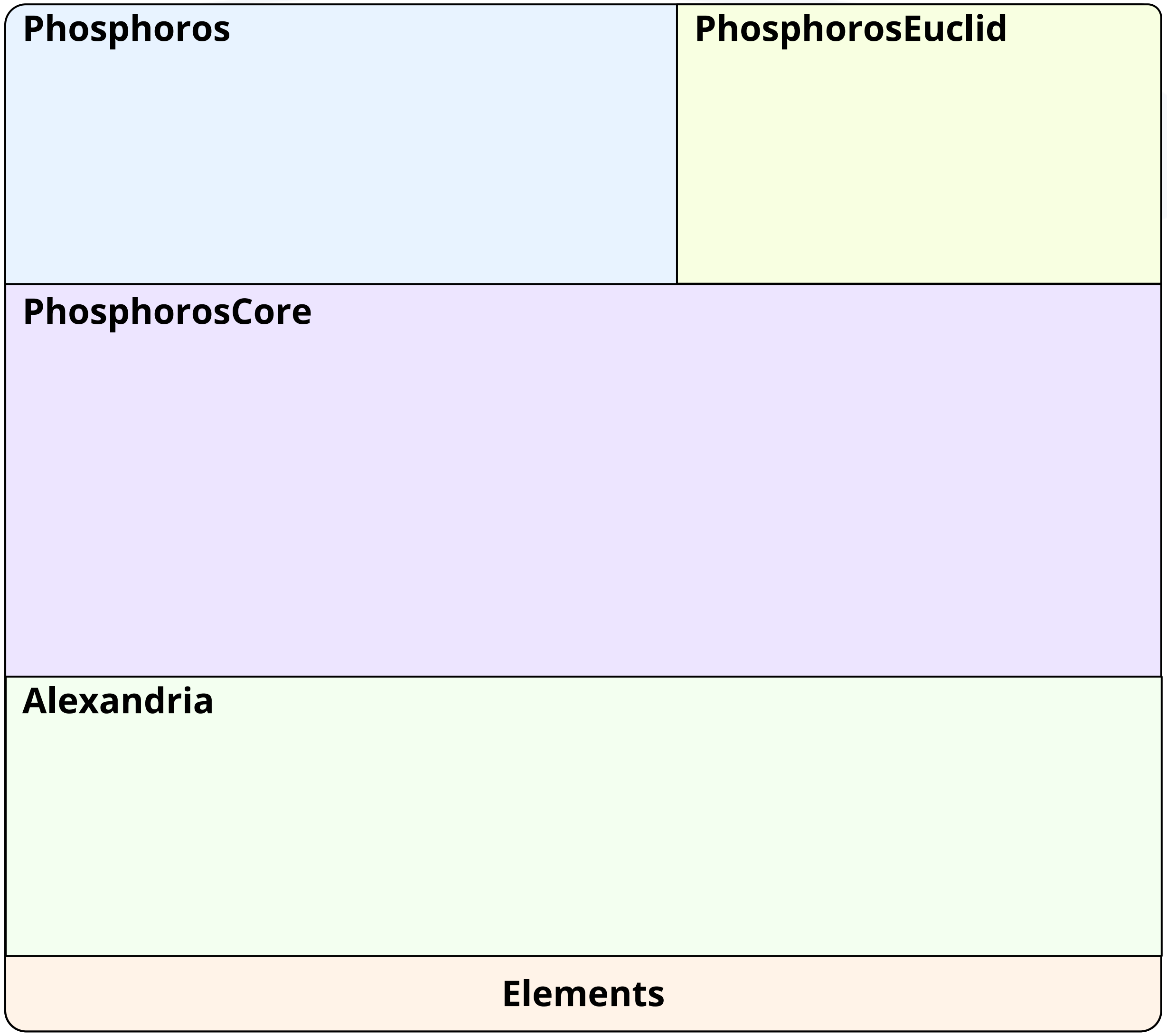
Project Organization



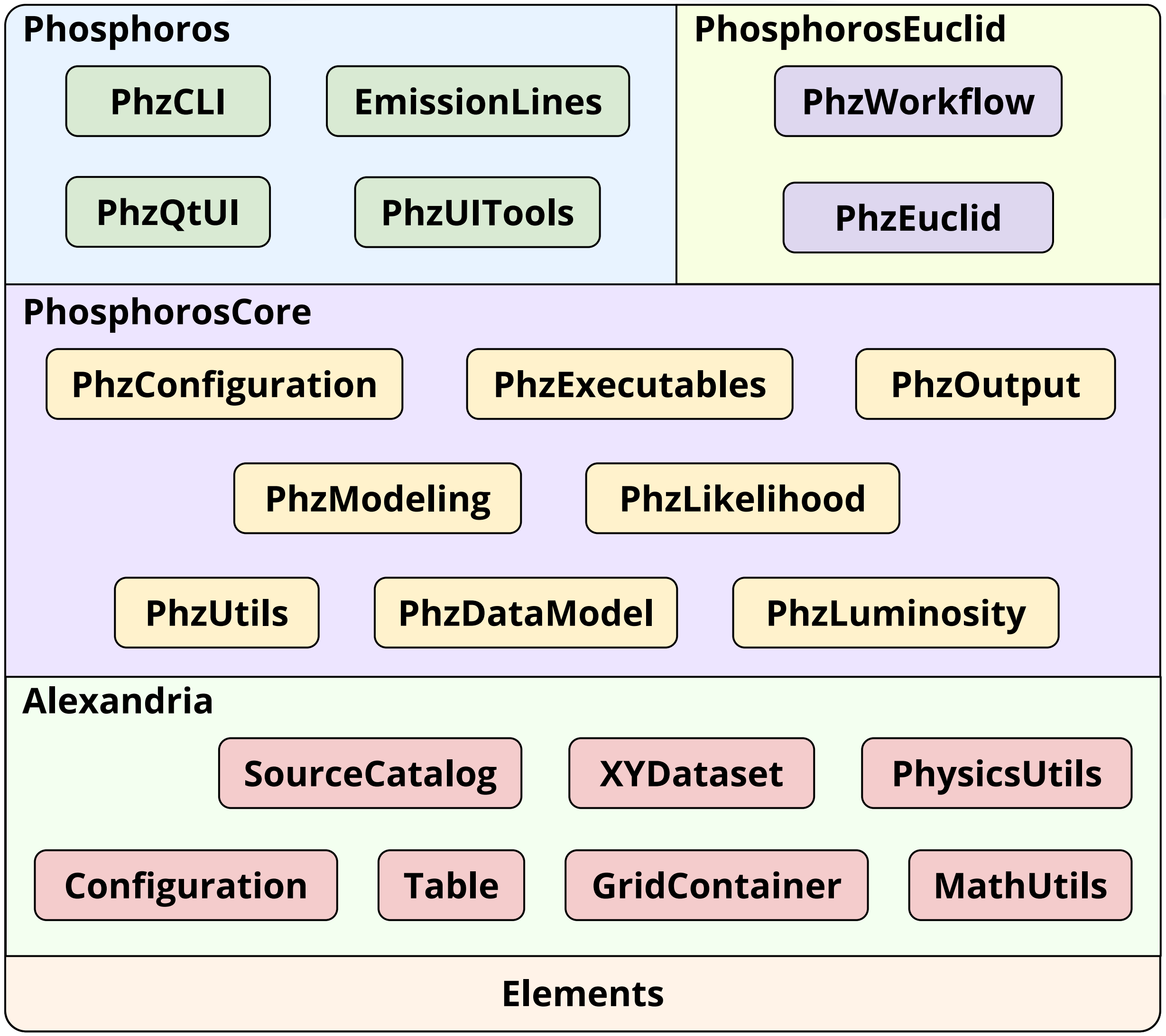
PhosphorosEuclid: PhosphorosCore wrapper for execution in Euclid ecosystem (see poster P3.5)

Software stack

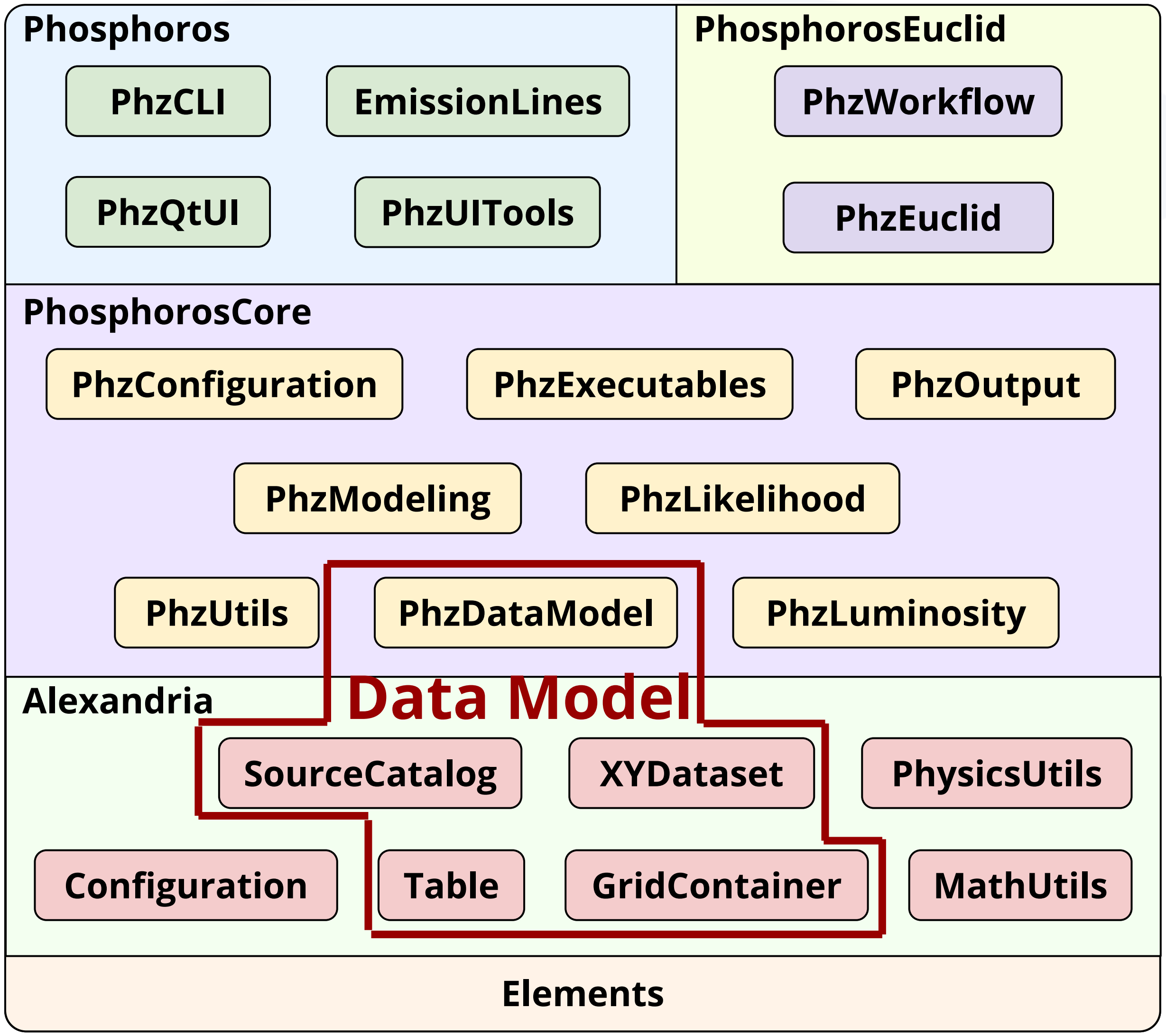
UNIVERSITÉ DE GENÈVE



Software stack



Software stack

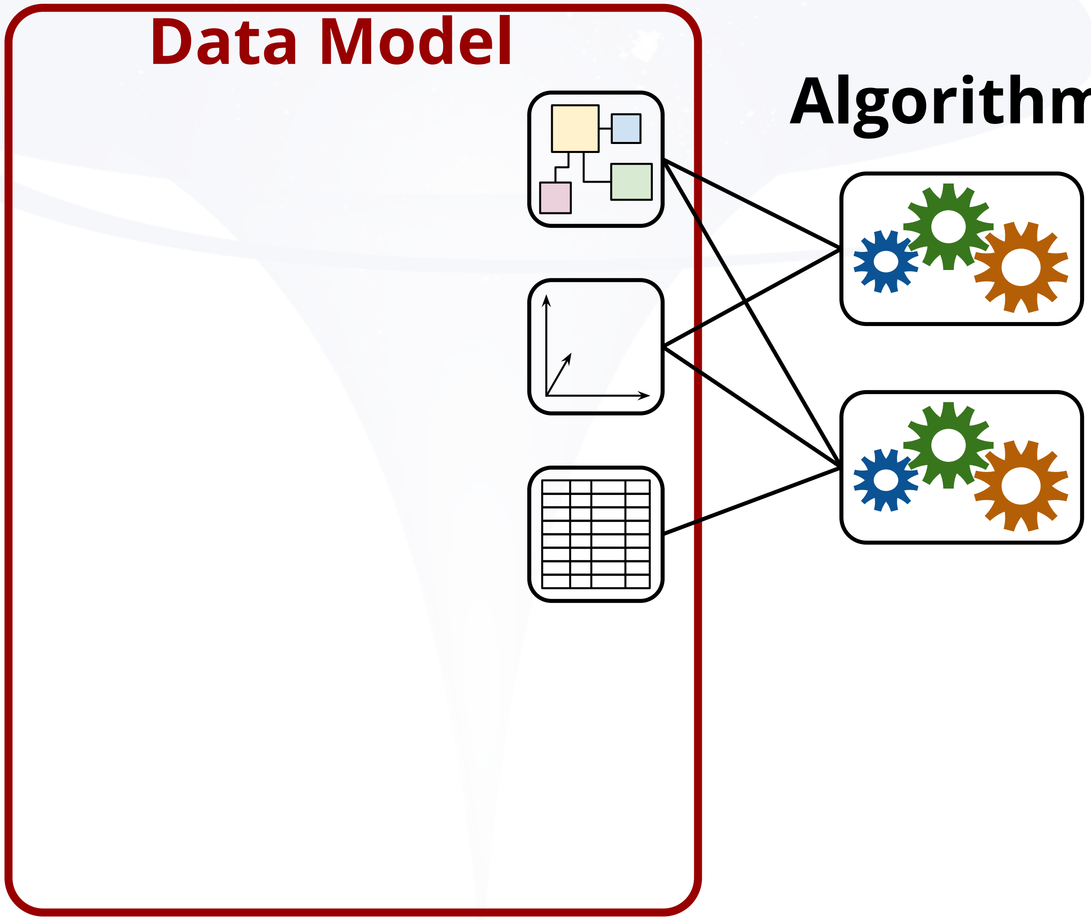


Data Model modules

Data Model modules

UNIVERSITÉ DE GENÈVE

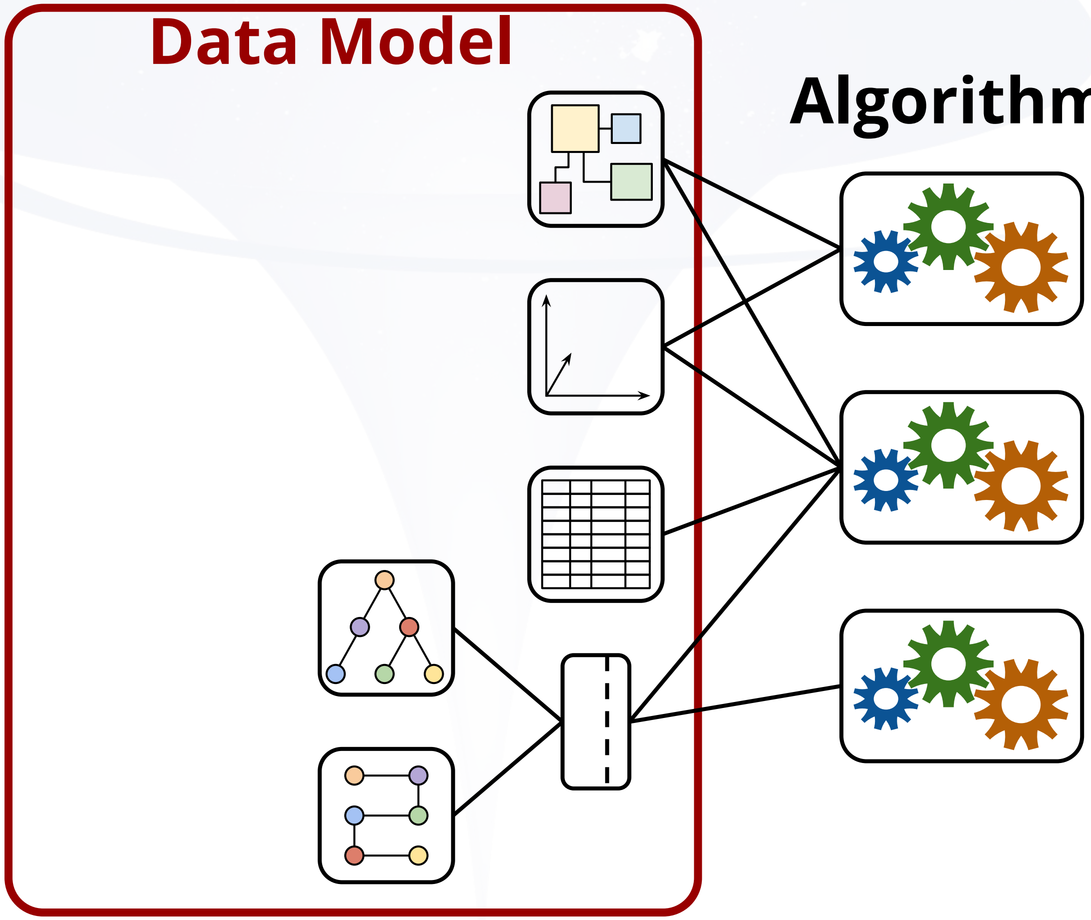
Define the data structures the algorithms use



Data Model modules

UNIVERSITÉ DE GENÈVE

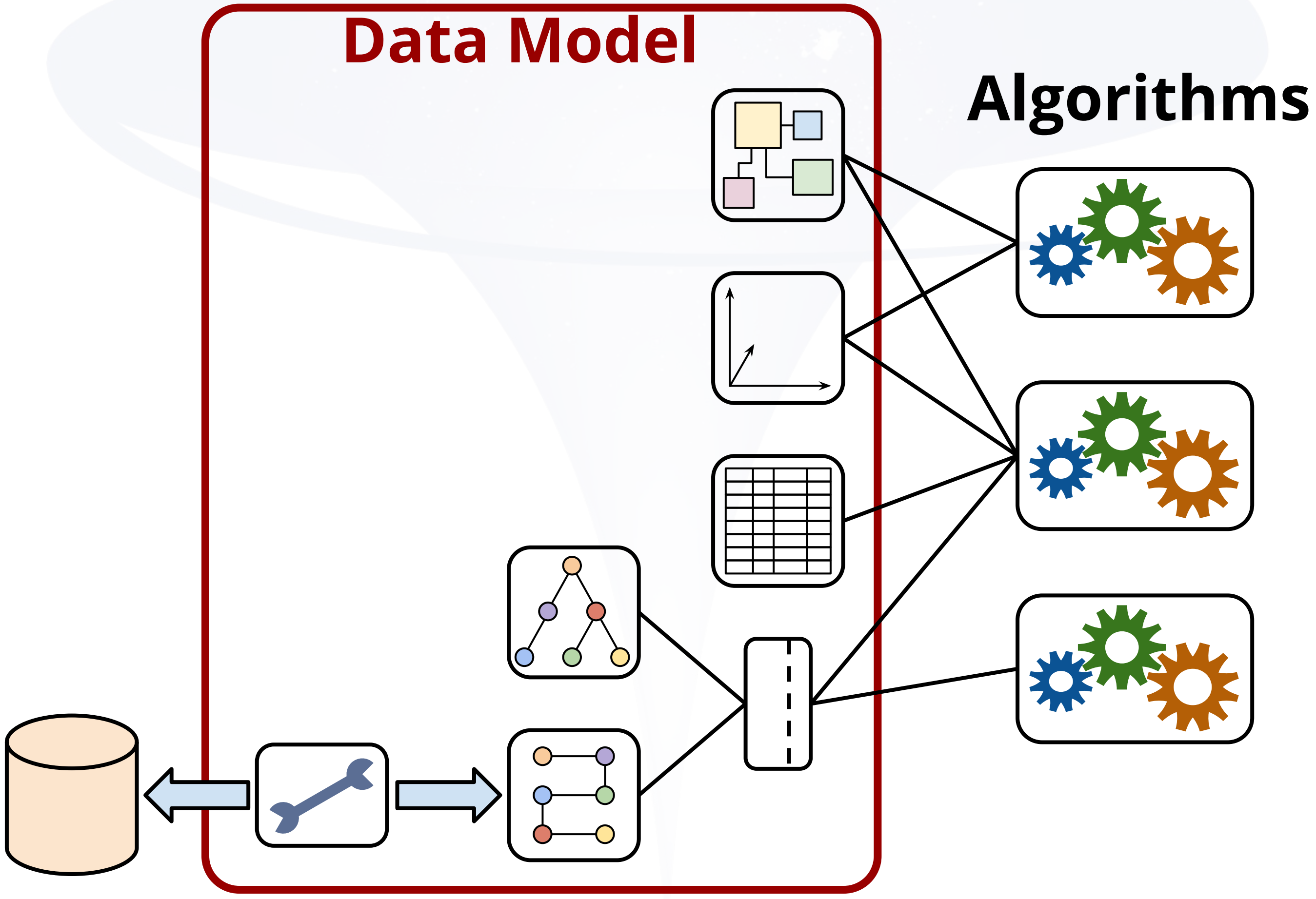
Hide specific implementations via interfaces



Data Model modules

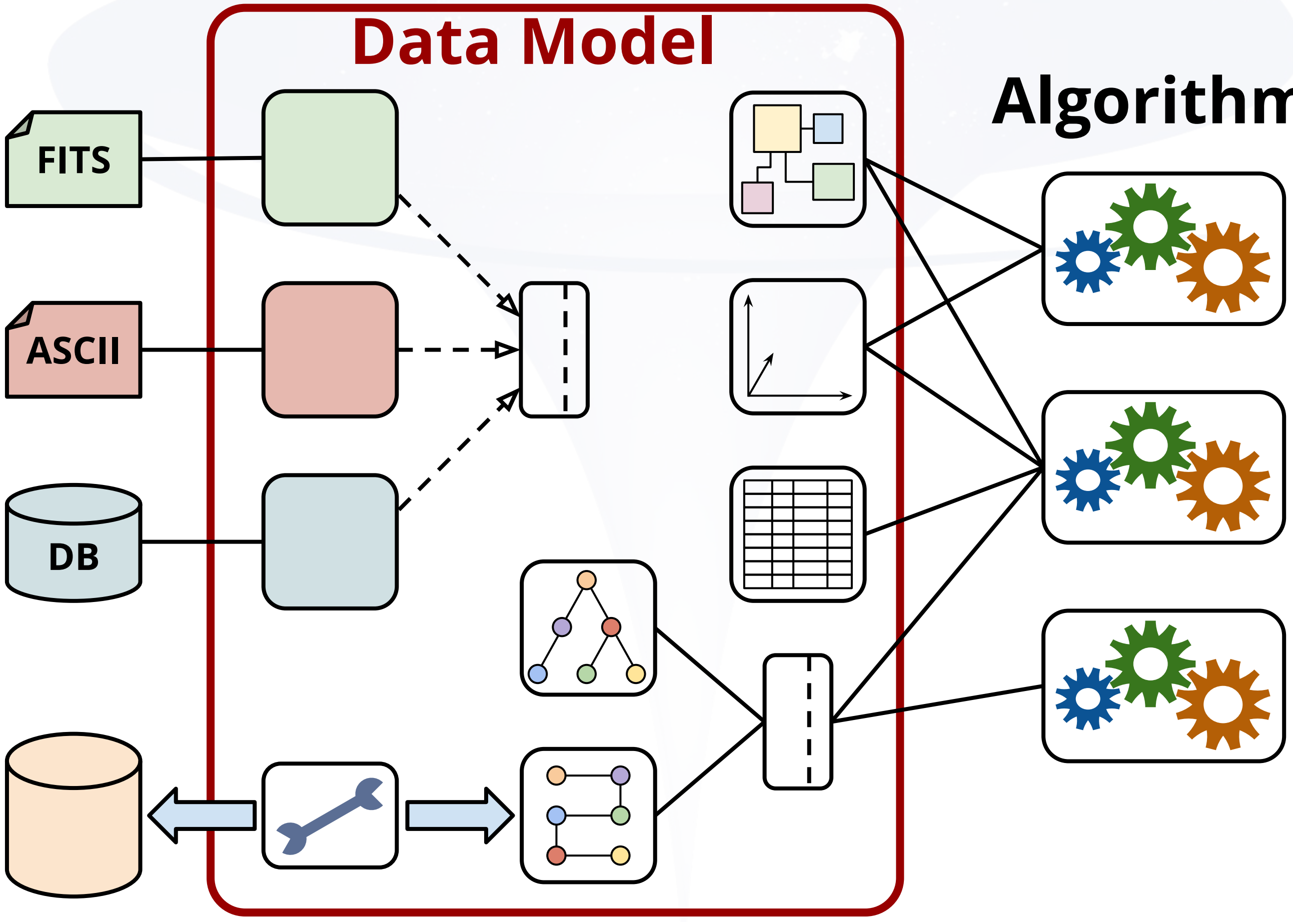
UNIVERSITÉ DE GENÈVE

Provide tools for serialization

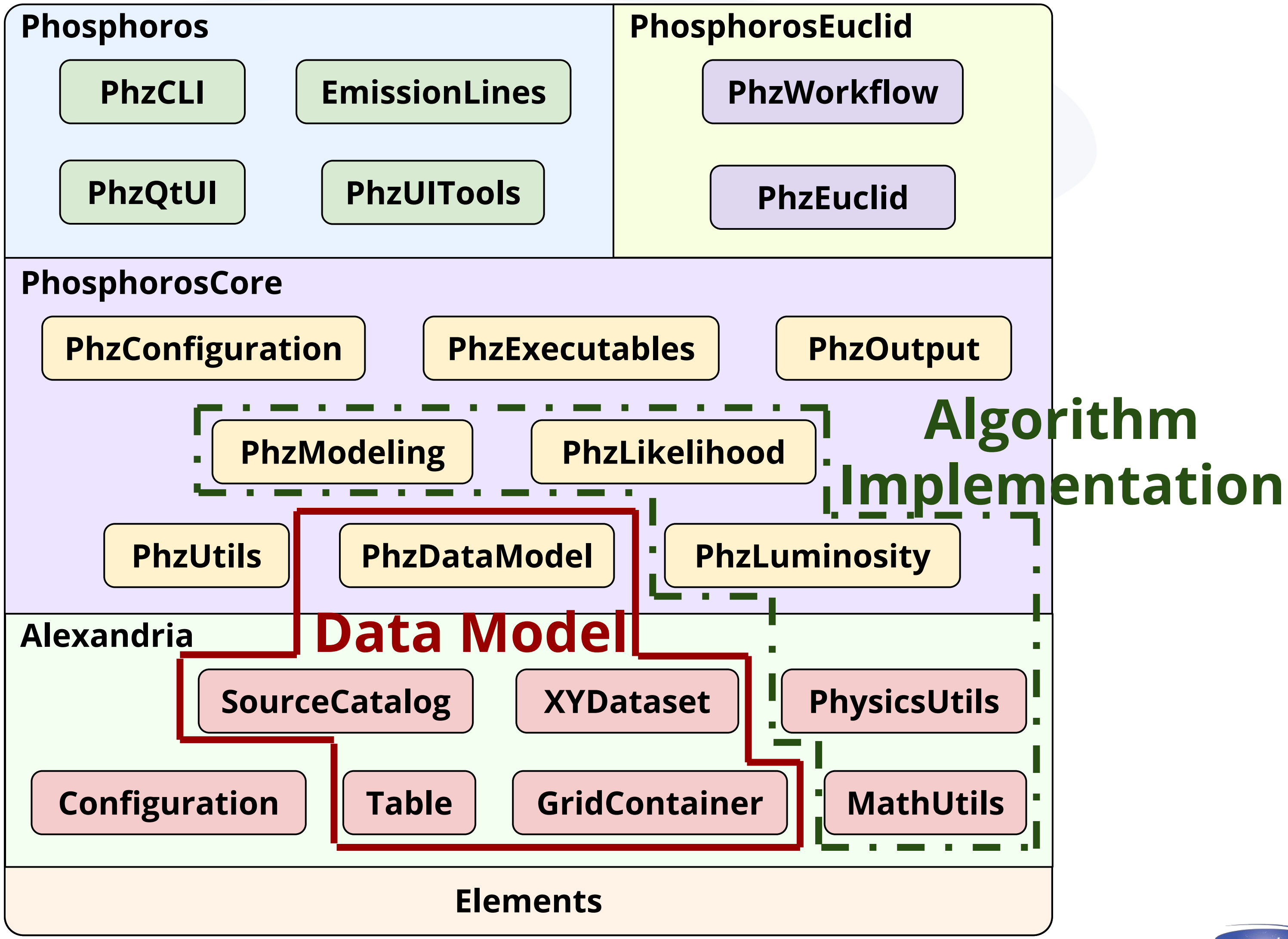


Data Model modules

Hide specific formats via abstraction



Software stack



Algorithm Impl modules

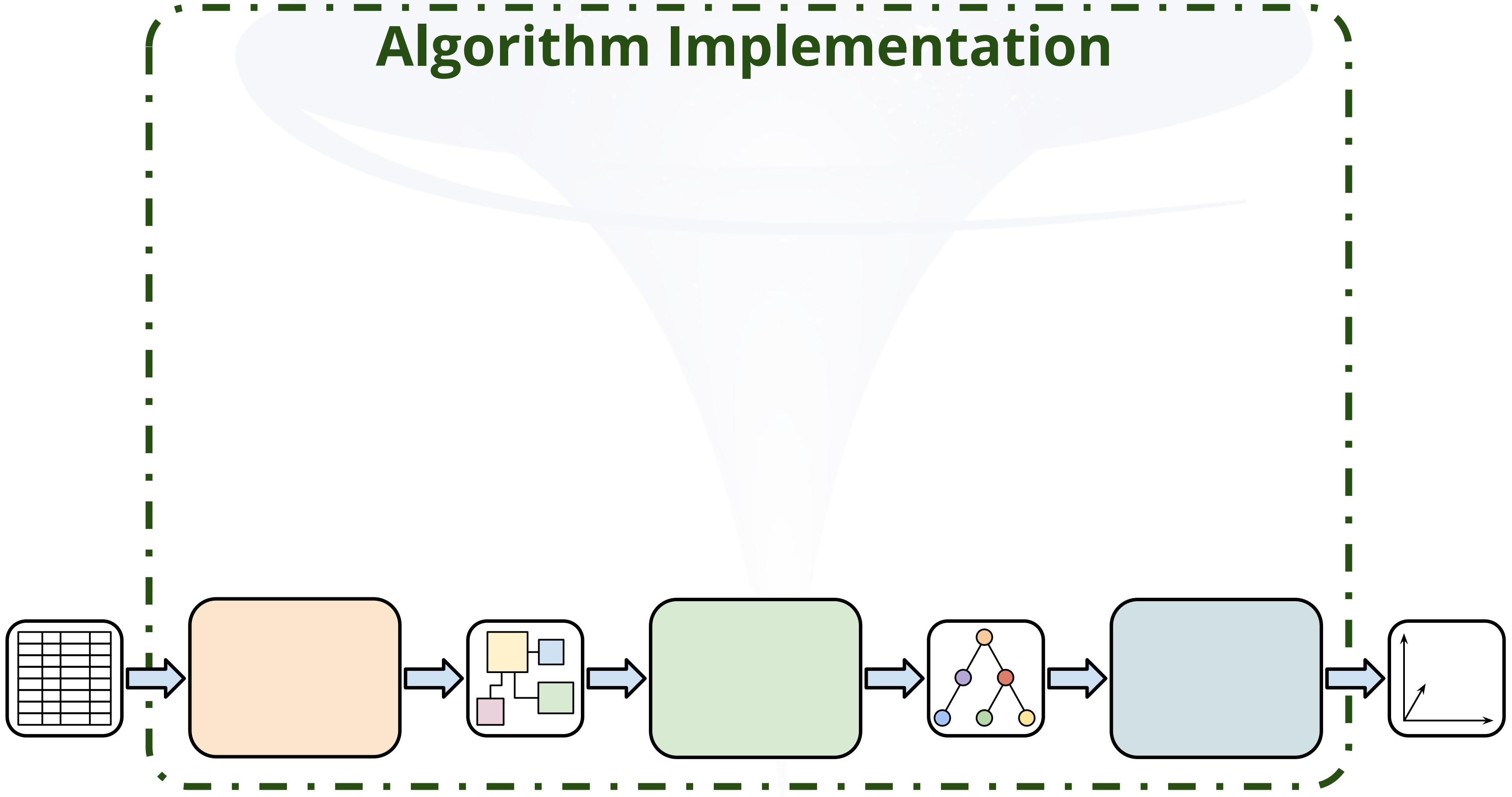
UNIVERSITÉ DE GENÈVE

Algorithm Implementation

Algorithm Impl modules

UNIVERSITÉ DE GENÈVE

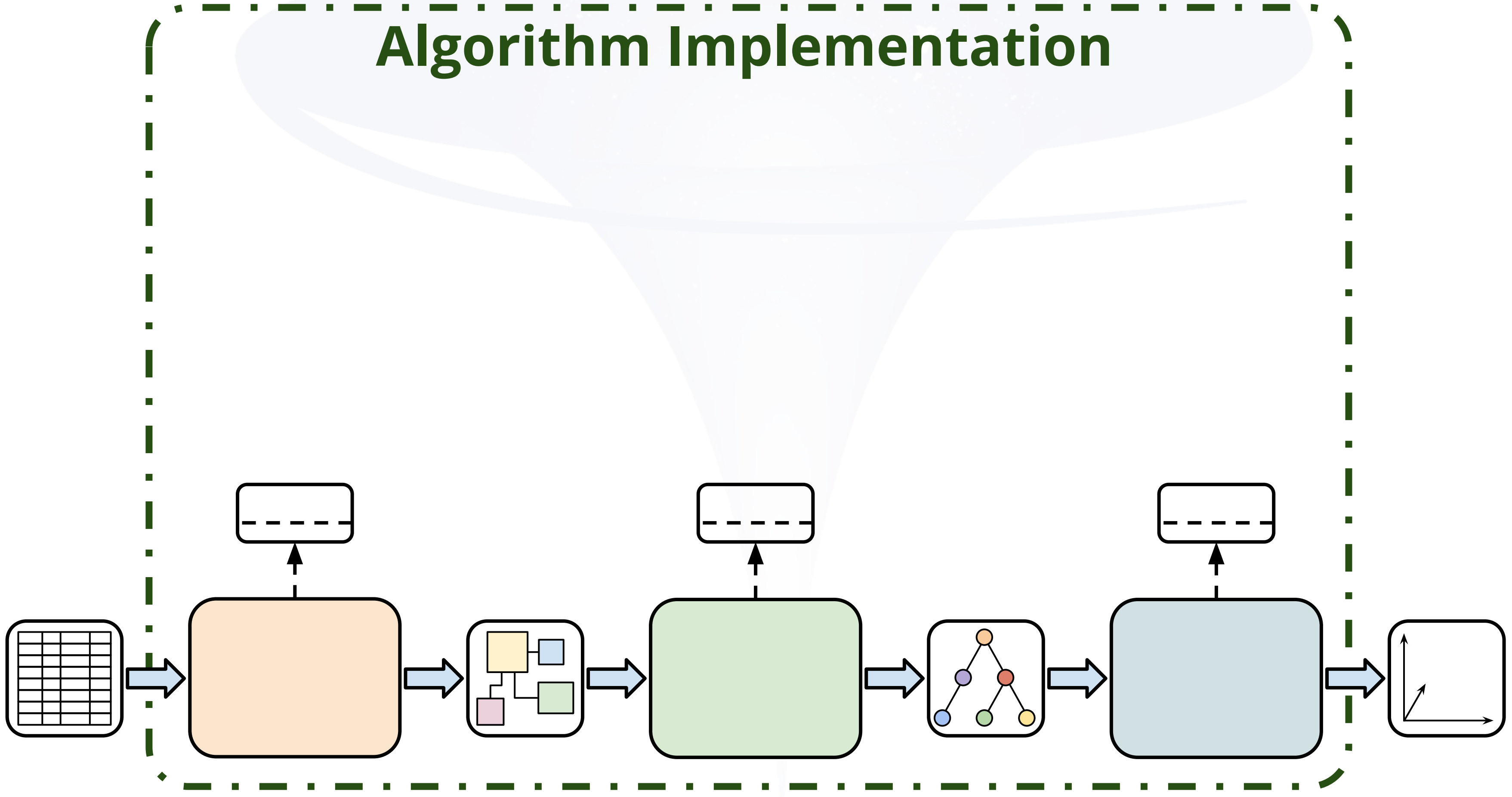
Organize the photo-z pipeline execution



Algorithm Impl modules

UNIVERSITÉ DE GENÈVE

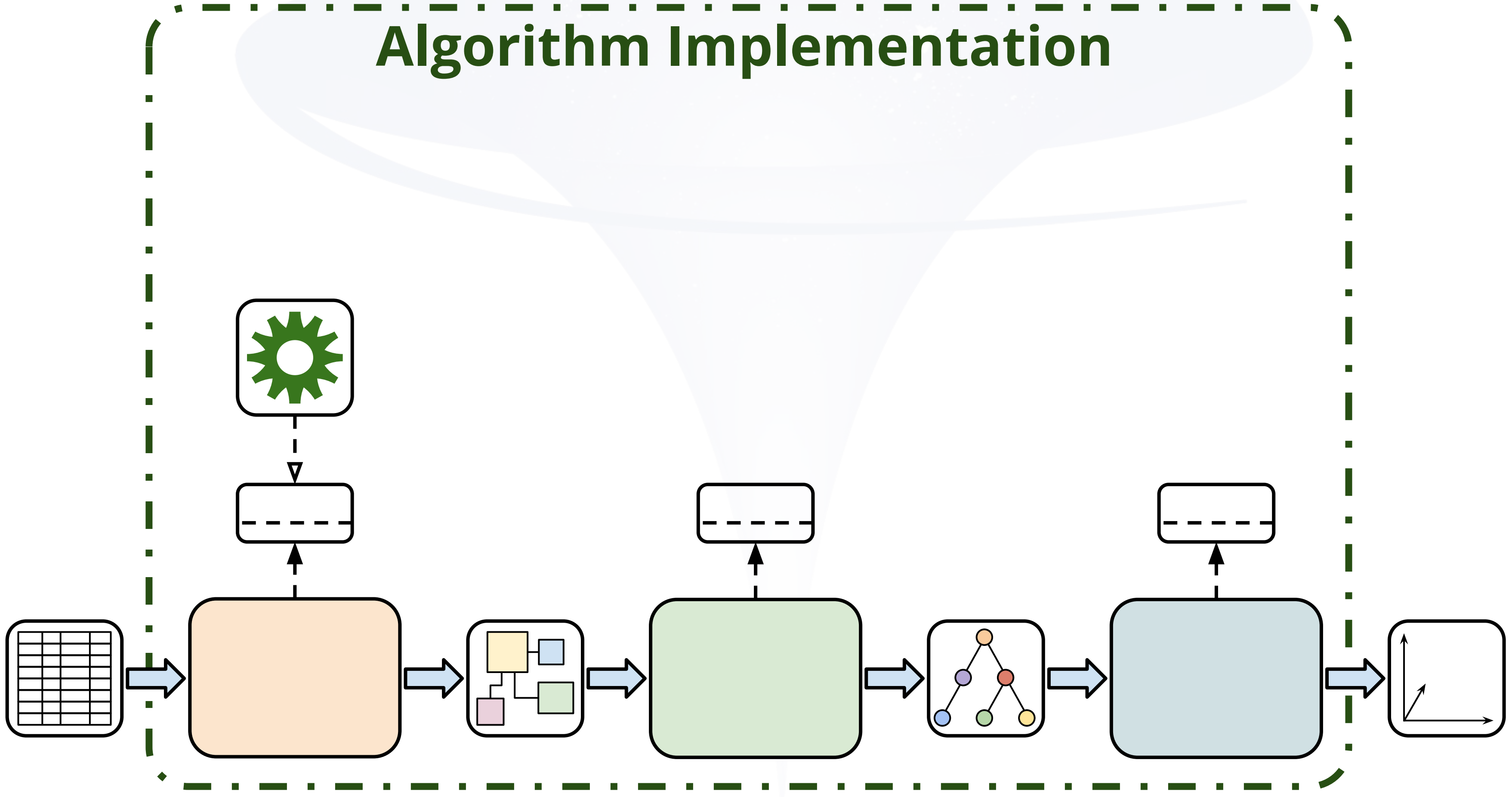
Abstracted steps as interfaces (std::function)



Algorithm Impl modules

UNIVERSITÉ DE GENÈVE

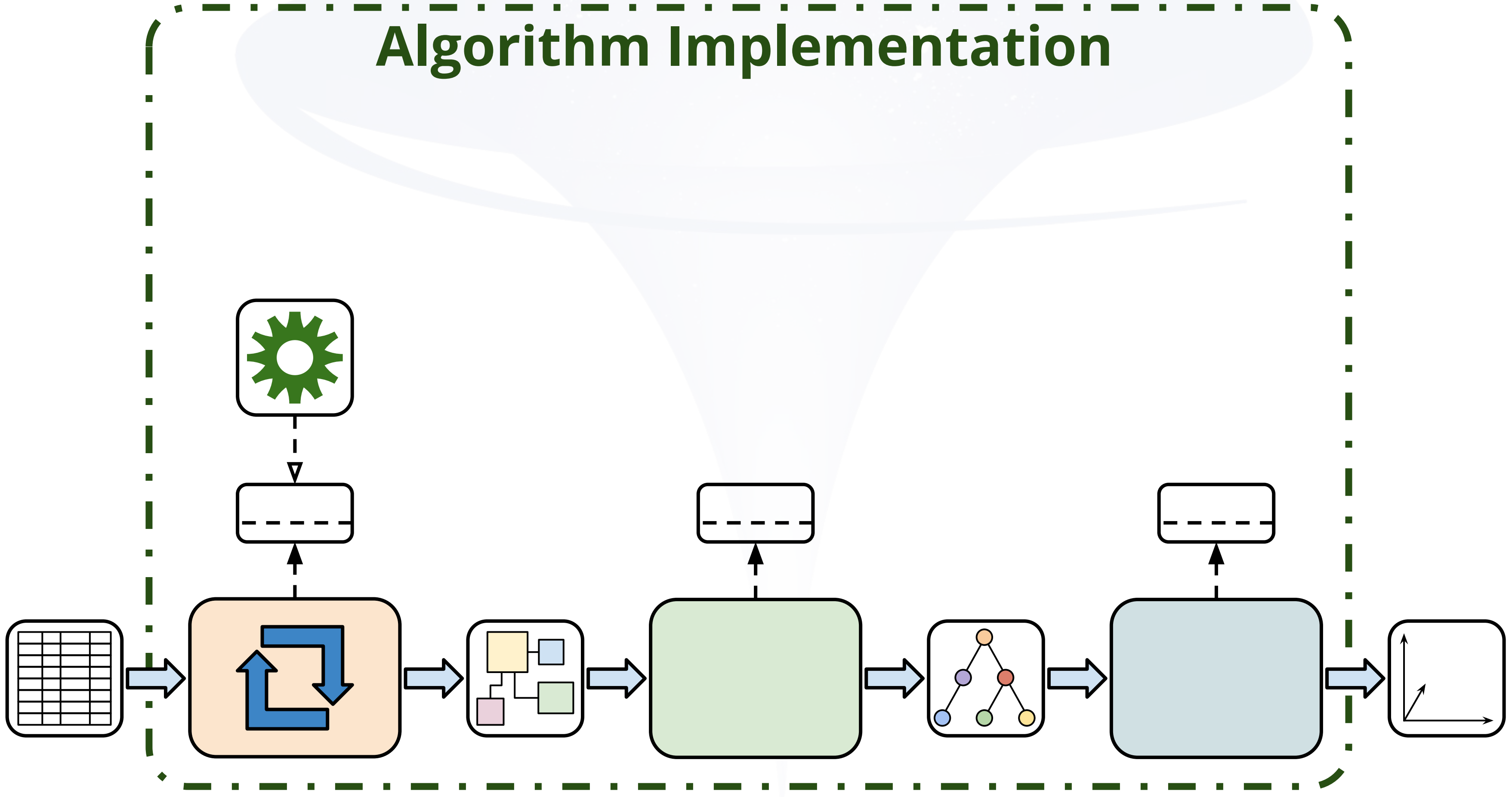
Logic implemented as reusable functors



Algorithm Impl modules

UNIVERSITÉ DE GENÈVE

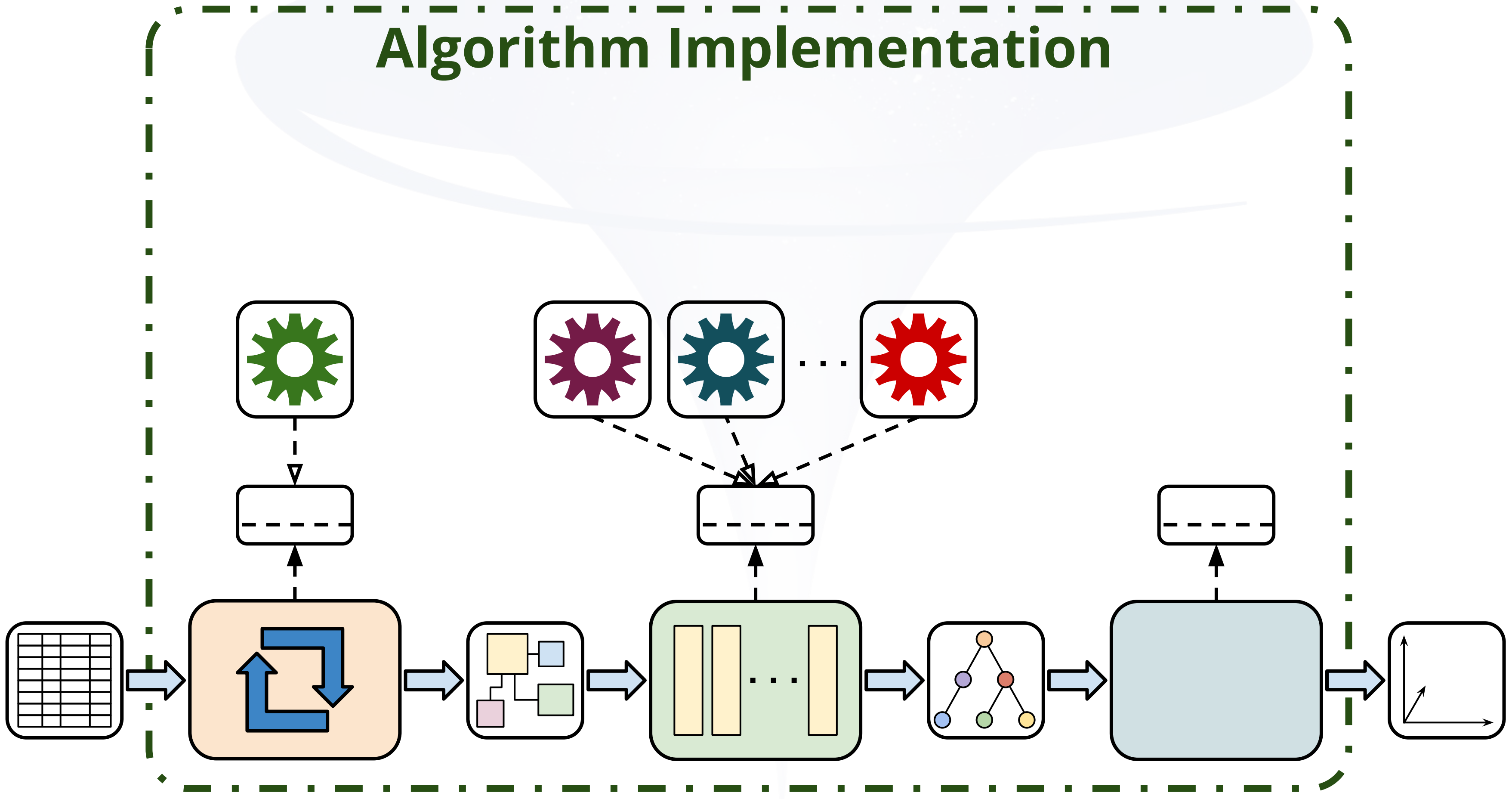
Loops are generalized using STL iterators



Algorithm Impl modules

UNIVERSITÉ DE GENÈVE

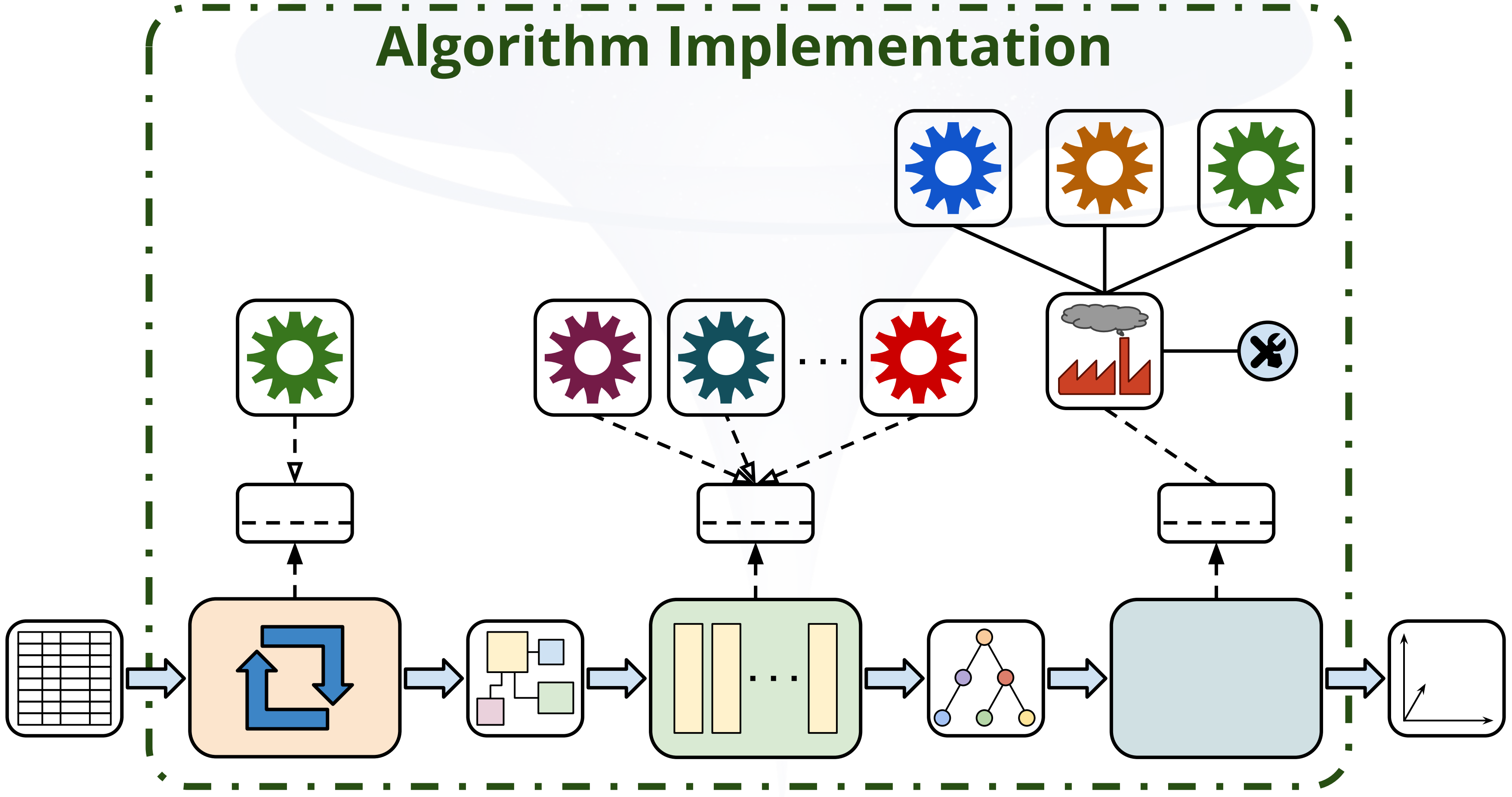
Composition is used to provide extensibility



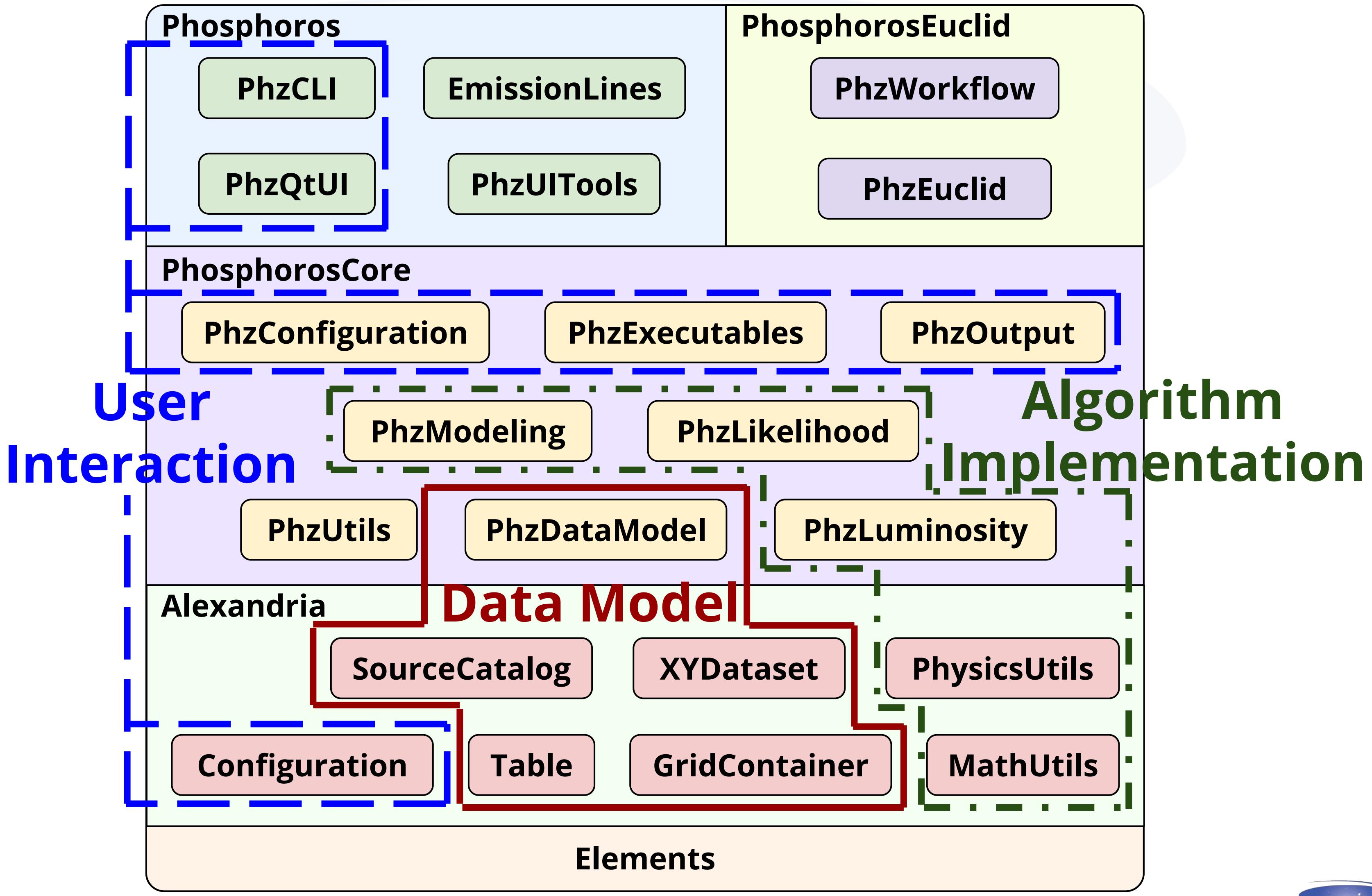
Algorithm Impl modules

UNIVERSITÉ DE GENÈVE

Factories are used for algorithm selection



Software stack

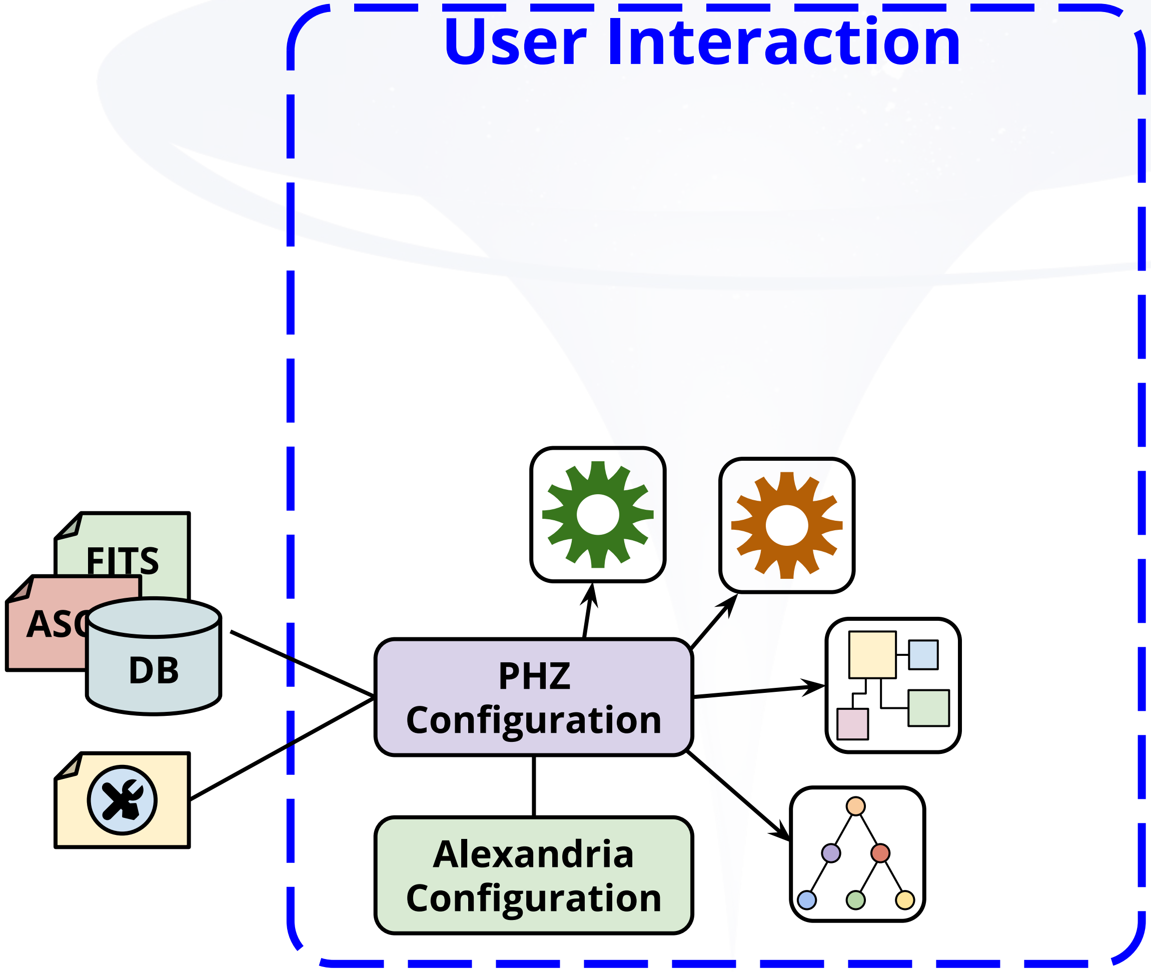


User Interaction modules

User Interaction modules

UNIVERSITÉ DE GENÈVE

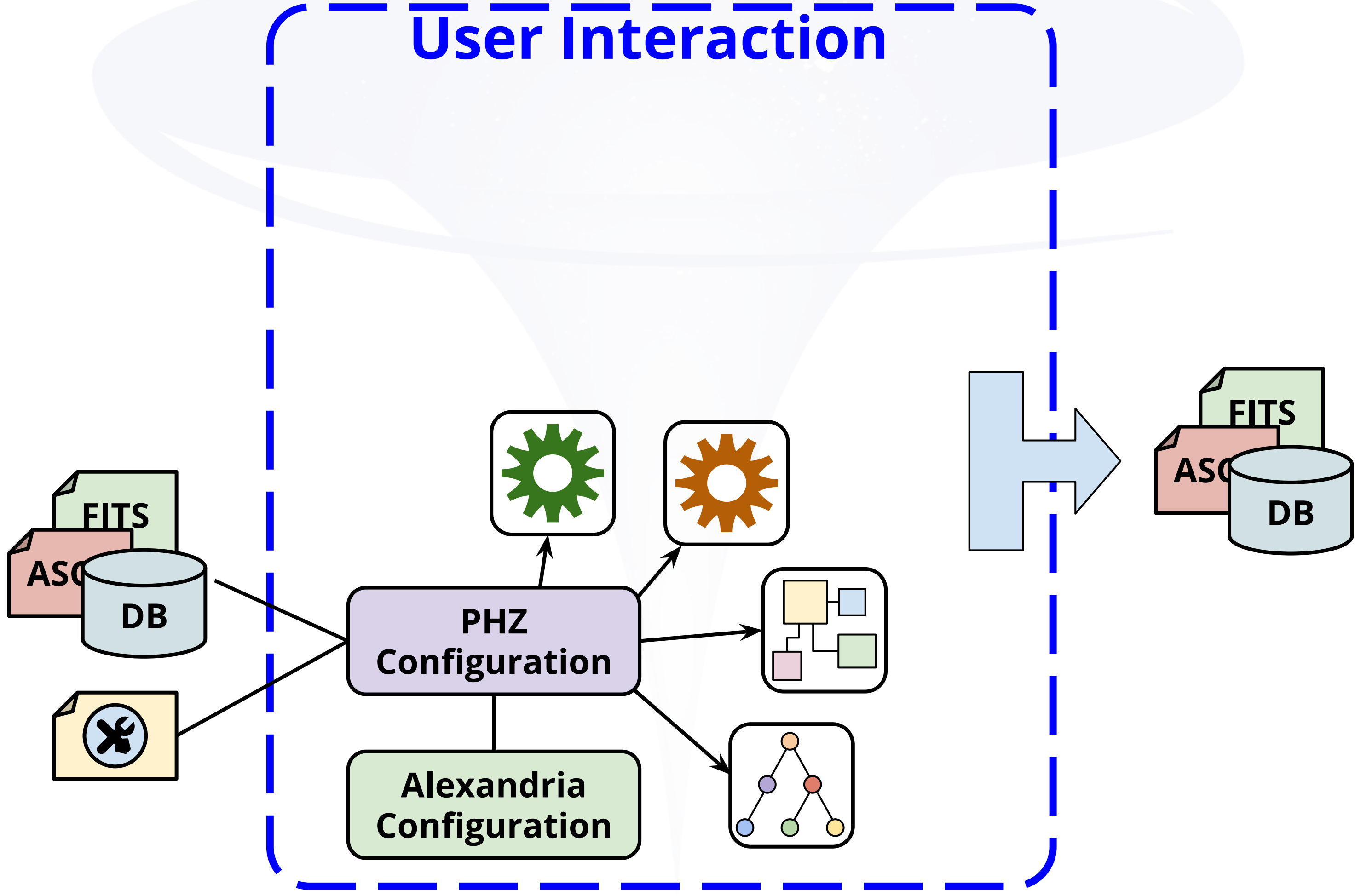
Configuration framework handles user input



User Interaction modules

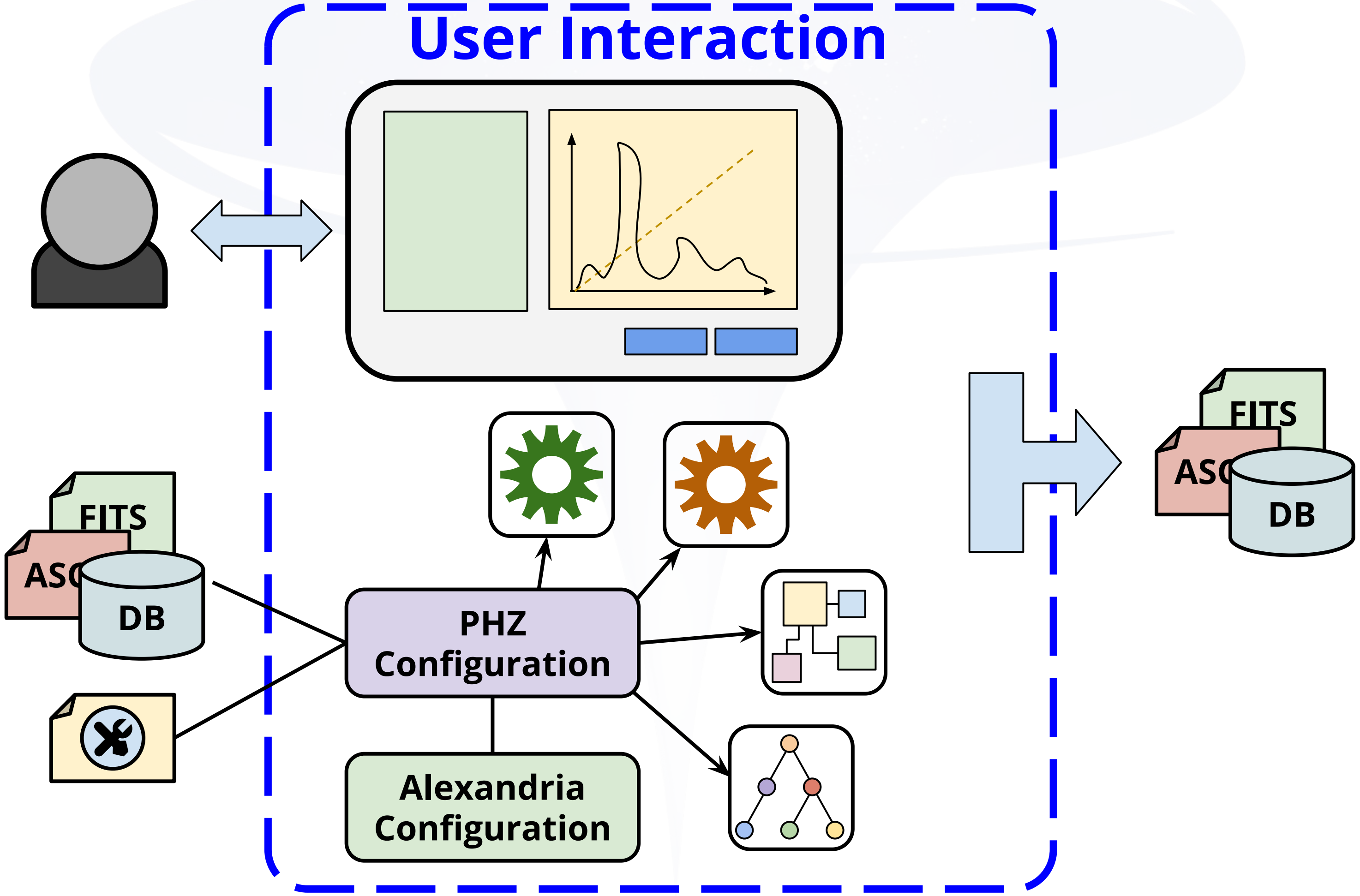
UNIVERSITÉ DE GENÈVE

Output generation

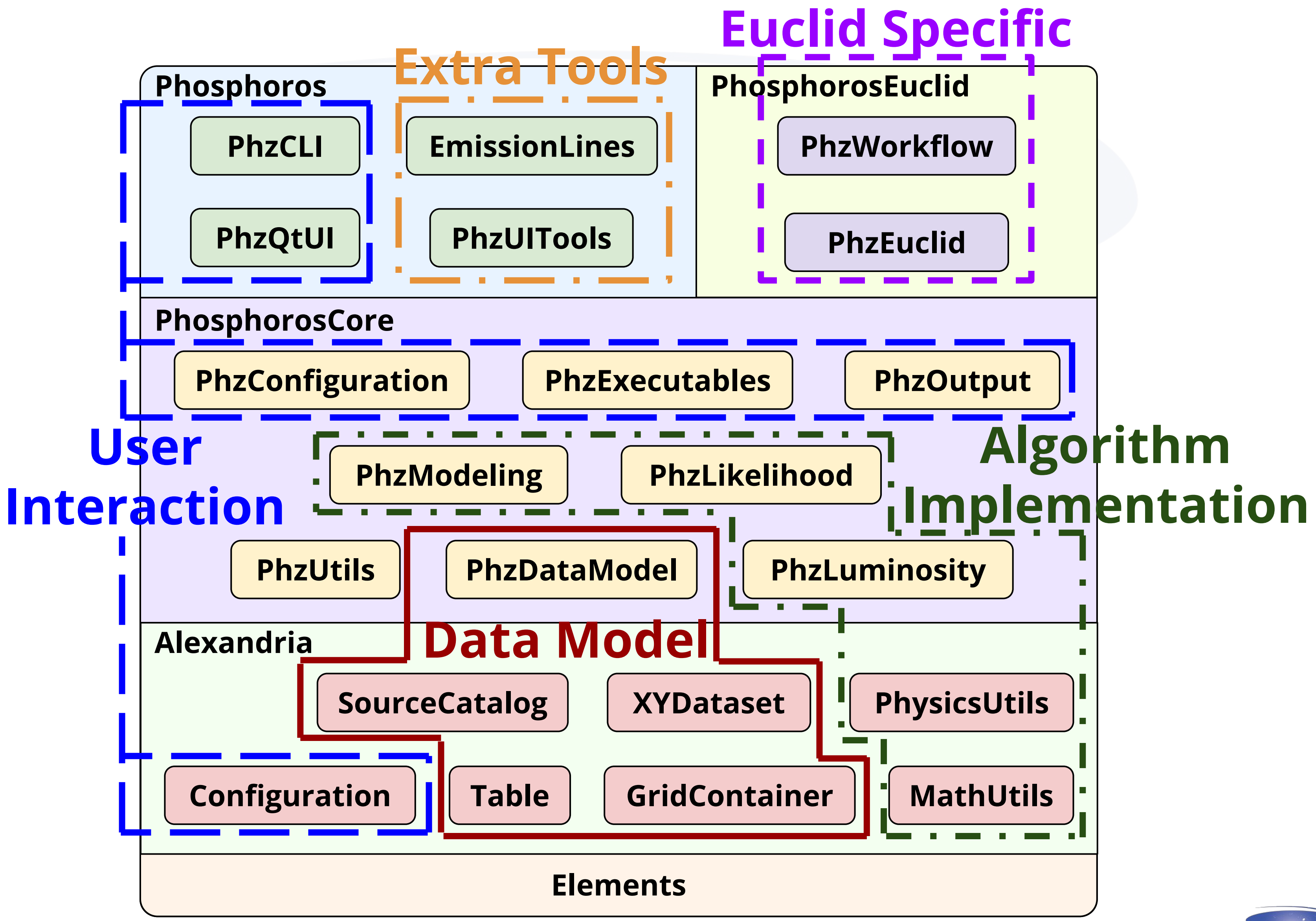


User Interaction modules

Graphical User Interface



Software stack



Current status

- Used for Euclid photo-z challenges
- Open source - licensed under LGPL
- Currently available only to Euclid Consortium
- Make public soon (Alexandria in 2016)
- Contacts for more info:
    - Nikolaos.Apostolakos@unige.ch
    - Pierre.Dubath@unige.ch