**Interactive Plot Visualization Tool**

**Overview**

This tool processes HDF5 sensor data files using configuration from XML and JSON files to generate interactive HTML visualizations. It supports various analysis visualizations including scatter plots, histograms, box plots, and statistical comparisons.

**Project Architecture**

The application follows a layered architecture pattern for separation of concerns:

1. **Main Module**: The main entry point of the application.
2. **Config Layer (XML/JSON)**: Handles parsing of XML configuration and JSON input/output mapping files.
3. **Persistence Layer (HDF Parser)**: Responsible for processing HDF5 data files.
4. **Data Storage**: Manages the storage and retrieval of processed data.
5. **Business Layer (Data Processing)**: Implements the core data processing and analysis logic.
6. **Presentation Layer (Plotly/HTML)**: Generates the interactive HTML reports with Plotly visualizations.

**Files and Directories**

**Configuration and Input Files**

* ConfigInteractivePlots.xml: XML configuration file for the application.
* InputsInteractivePlot.json: JSON configuration file for all sensor inputs and outputs.
* InputsPerSensorInteractivePlot.json: JSON configuration file for per-sensor input/output processing.

**Main Application Package**

* InteractivePlot/: Contains the main application modules:
  + a\_config\_layer/: Modules for parsing XML and JSON configuration files.
  + b\_persistence\_layer/: Modules for processing HDF5 data files.
  + c\_data\_storage/: Modules for managing data storage and models.
  + d\_business\_layer/: Modules for implementing the core data processing and analysis logic.
  + e\_presentation\_layer/: Modules for generating interactive HTML reports with Plotly visualizations.

**KPI Module**

* KPI/: Contains modules for calculating and visualizing Key Performance Indicators (KPIs).
  + a\_business\_layer/: Modules for KPI calculation logic.
  + b\_presentation\_layer/: Modules for KPI visualization.

**Testing**

* tests/: Contains unit tests for the project.

**Documentation and Diagrams**

* docs/: Contains documentation and diagrams for the project architecture and design.

**Running the Application**

To run the application, use the following command:

python main.py ConfigInteractivePlots.xml InputsInteractivePlot.json

This command will:

1. Parse the XML configuration file using the XmlConfigParser.
2. Determine the HDF5 file type.
3. Use the JSONParserFactory to obtain the appropriate JSON parser and generate an input/output map.
4. Process the HDF5 files using the HdfProcessorFactory.

**Testing**

Unit tests are provided in the tests/ directory. To run all the tests, use your preferred Python testing framework (e.g., pytest):

pytest tests/

Make sure to install pytest if you haven't already:

pip install pytest

**Development Tools**

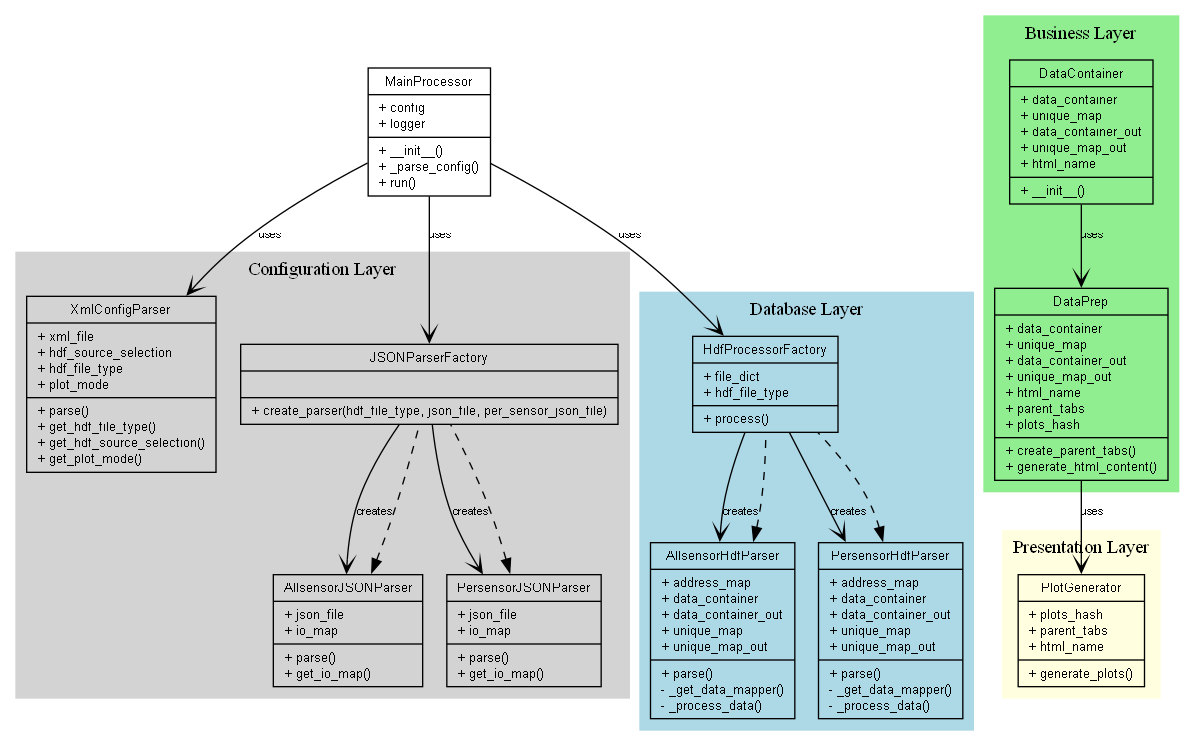
* Visual Studio Code is configured using files in .vscode/.
* Diagrams and documentation images can be found in the docs/ folder for an overview of the project architecture.

**Additional Information**

* The project uses a layered approach to separate configuration parsing, persistence, business logic, and presentation.
* The main entry point of the application is main.py.

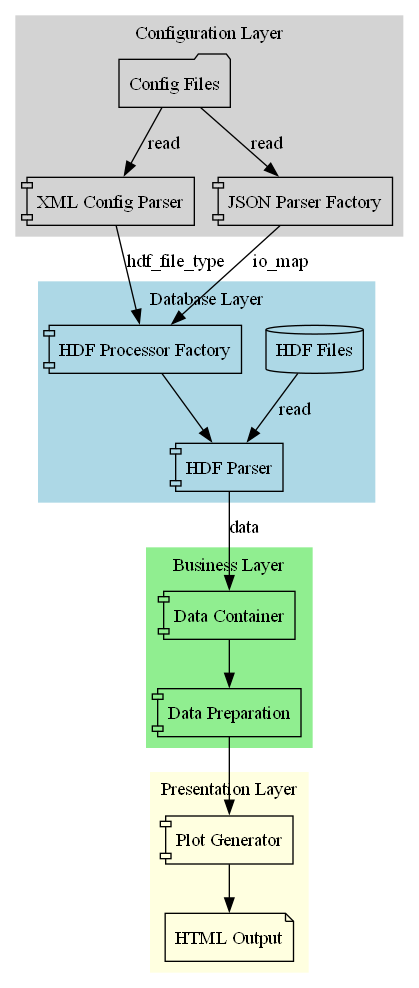
Class Diagram

The class diagram represents the object-oriented design of the Interactive Plot Visualization Tool. It shows the various classes, their attributes, and the relationships between them. This diagram helps to understand the structure and organization of the codebase, making it easier to maintain and extend the application.



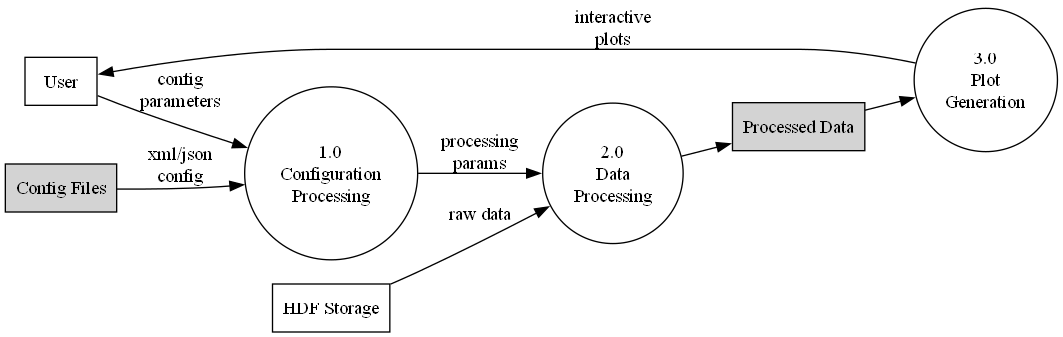
Component Diagram:

The component diagram depicts the high-level components of the Interactive Plot Visualization Tool and their interactions. It provides an overview of the major subsystems or modules, such as the Config Layer, Persistence Layer, Data Storage, Business Layer, and Presentation Layer. This diagram helps to visualize the overall system architecture and the dependencies between the components.



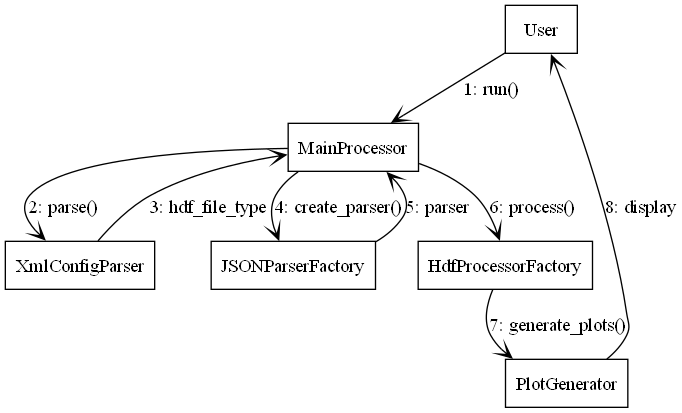
Data Flow Diagram:

The data flow diagram illustrates the flow of data through the different components of the Interactive Plot Visualization Tool. It shows how the data is processed, transformed, and passed between the various modules, such as the HDF5 file processing, data storage, and visualization generation. This diagram helps to understand the data processing pipeline and the transformations that occur at each stage.



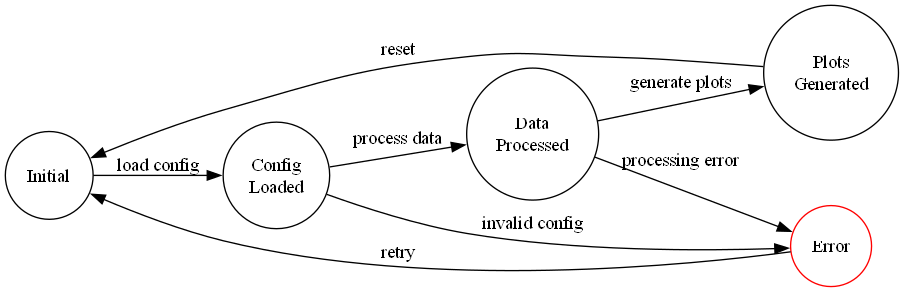
Sequence Diagram:

The sequence diagram depicts the dynamic interactions between the different objects or components of the Interactive Plot Visualization Tool. It shows the sequence of messages or method calls exchanged between the objects to achieve a specific functionality, such as processing HDF5 files, generating visualizations, or producing HTML reports. This diagram helps to understand the flow of control and the collaboration between the different parts of the system.



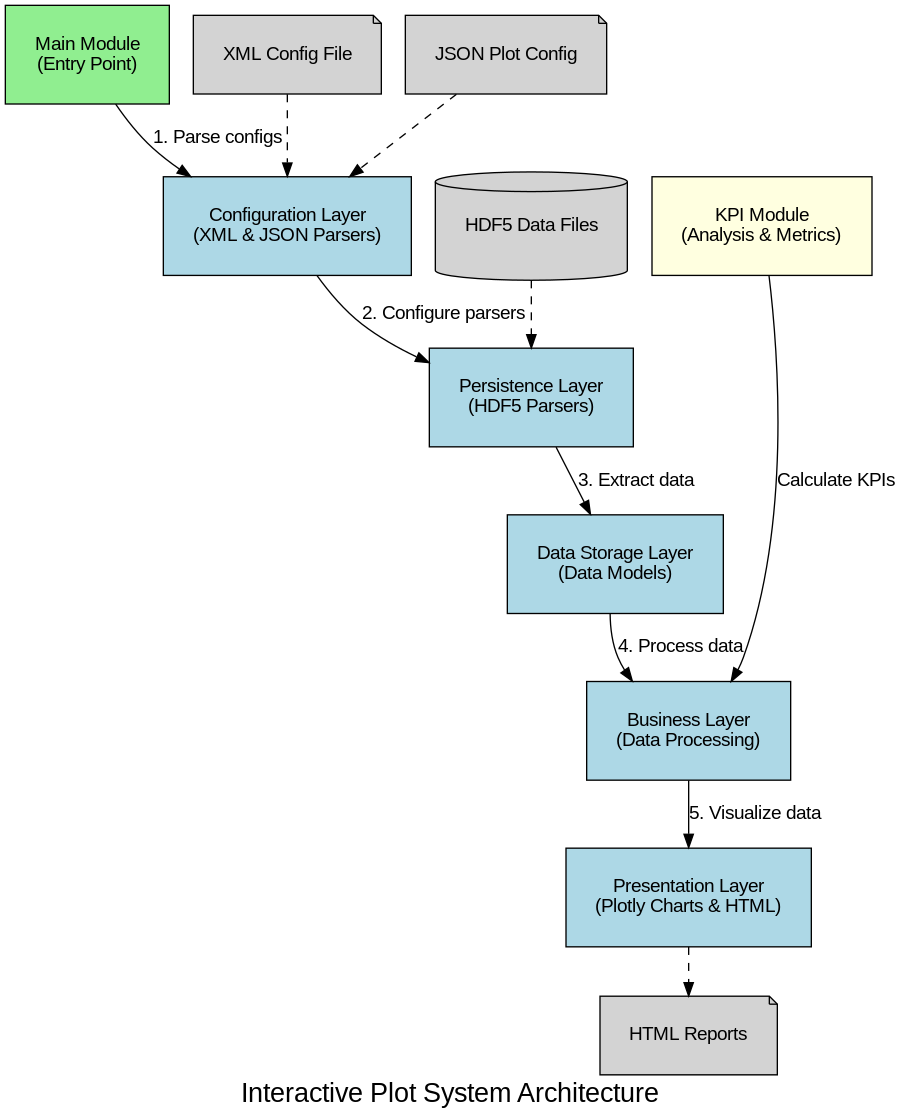
State Machine Diagram :

The state machine diagram represents the different states and transitions of the Interactive Plot Visualization Tool. It models the behavior of the system, showing how it responds to various events or user actions. This diagram can be useful for understanding the different states the application can be in, such as processing data, generating visualizations, or handling errors, and how it transitions between these states.



System Overview :

The system overview diagram provides a high-level view of the Interactive Plot Visualization Tool and its main components. It shows the overall architecture, including the layered structure with the Config Layer, Persistence Layer, Data Storage, Business Layer, and Presentation Layer. This diagram helps to understand the big picture of the system and the relationships between the major subsystems.



Html generation part

