

# **2D LIDAR MAPPING**

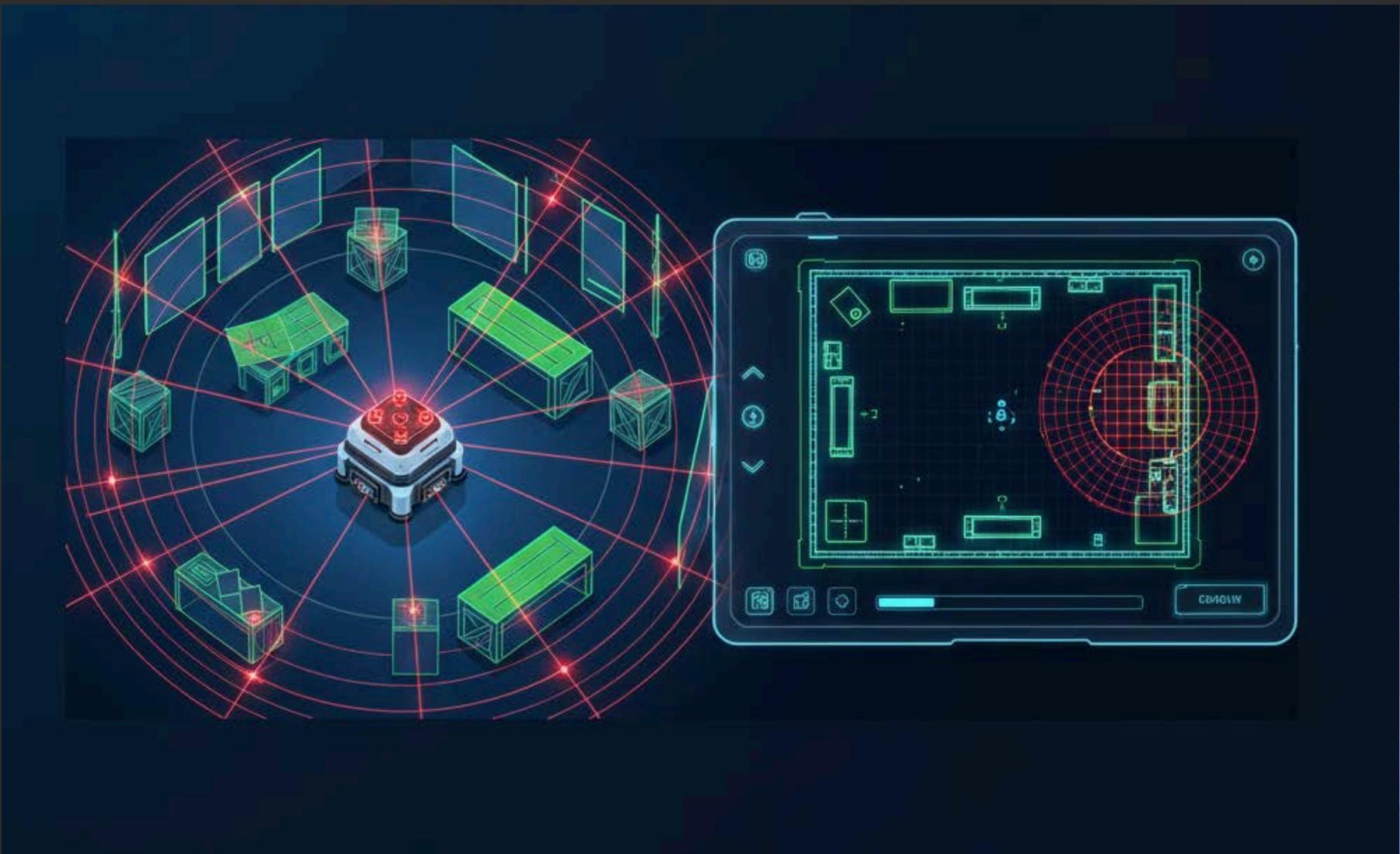
**LiStars**

Presented by

**Zac Swisher, Sneha Patel, Jamil Velez, Bryton Montgomery**

# PROJECT OVERVIEW

- A room-mapping system using LiDAR sensors to detect and map obstacles in real time.
- Data is processed and displayed on a 2D graphical user interface (GUI) as an overhead "map" of the environment.



# PROJECT OVERVIEW

- **Initial Phase:** The system operates as a stationary or manually moved platform.
- **Later Iterations:** The platform will be motorized to autonomously navigate and continuously update the map.

# 3 MAIN FEATURES

1

**Real-Time LiDAR Data Acquisition and Processing :** Continuously captures distance measurements from the LiDAR sensor and processes them into usable coordinate data. Enables instant obstacle detection and mapping for dynamic environments.

2

**2D Map Visualization GUI:** Provides a real-time graphical interface to display scanned environments. Allows users to view, interpret, and interact with obstacle maps easily.

3

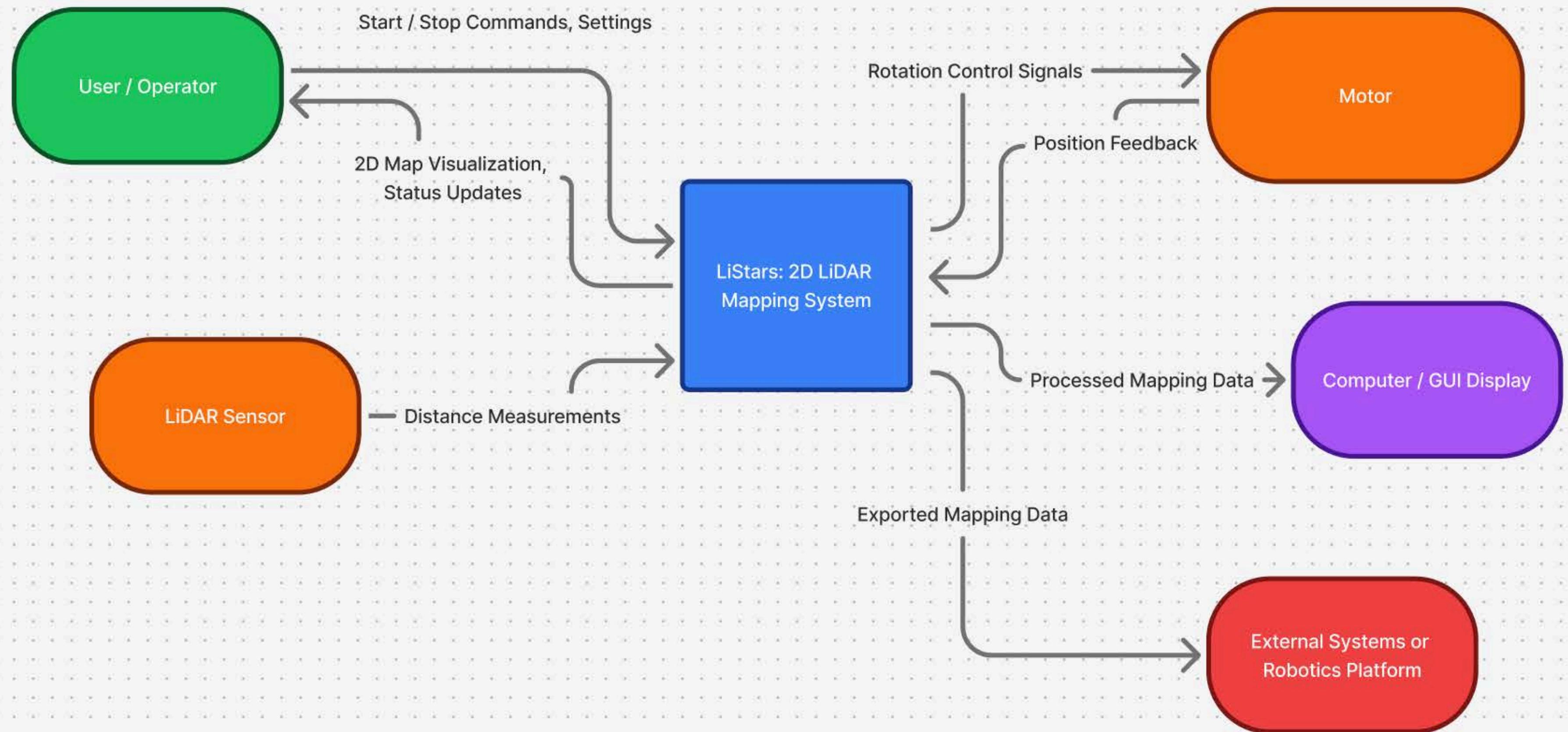
**Compatibility with external systems:** Enables seamless data sharing with external platforms through standardized formats and protocols. Ensures integration with robotics frameworks, research tools, and custom applications.

# CONTEXT DIAGRAM

**Purpose:** Provides a high-level overview of the entire LiStars system and how it interacts with external entities such as the User/Operator, LiDAR Sensor, Motor, Computer Display, and External Systems.

## Key Features Represented:

- The User controls and monitors the mapping process.
- The LiDAR Sensor sends distance data to the system.
- The Motor receives control signals for rotation.
- The Computer GUI displays processed maps.
- External systems can use exported map data for further automation or navigation.



# DATA FLOW DIAGRAM

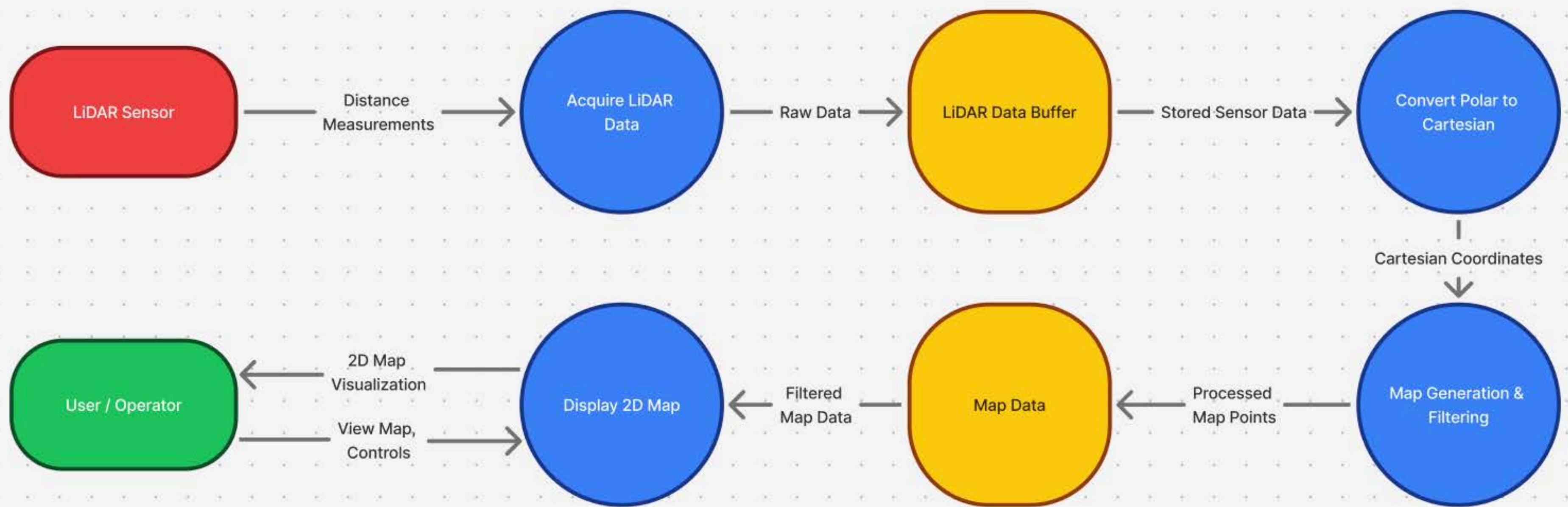
**Purpose:** Zooms into the data processing pipeline — showing how LiDAR measurements are transformed step by step into visual map data.

## Key Features Represented:

- Acquire LiDAR Data from the sensor.
- Convert Polar to Cartesian coordinates for mapping.
- Filter and Generate Map to clean and process data.
- Display 2D Map in the GUI for the user.
- The User/Operator interacts by viewing and controlling map visualization.

# DATA FLOW DIAGRAM

**Relation to Context Diagram:** This Data flow Diagram zooms in on the flow between the LiDAR Sensor and User/Operator identified in the Context Diagram, detailing how raw sensor data becomes the 2D map the user sees.



# USE CASE DIAGRAM

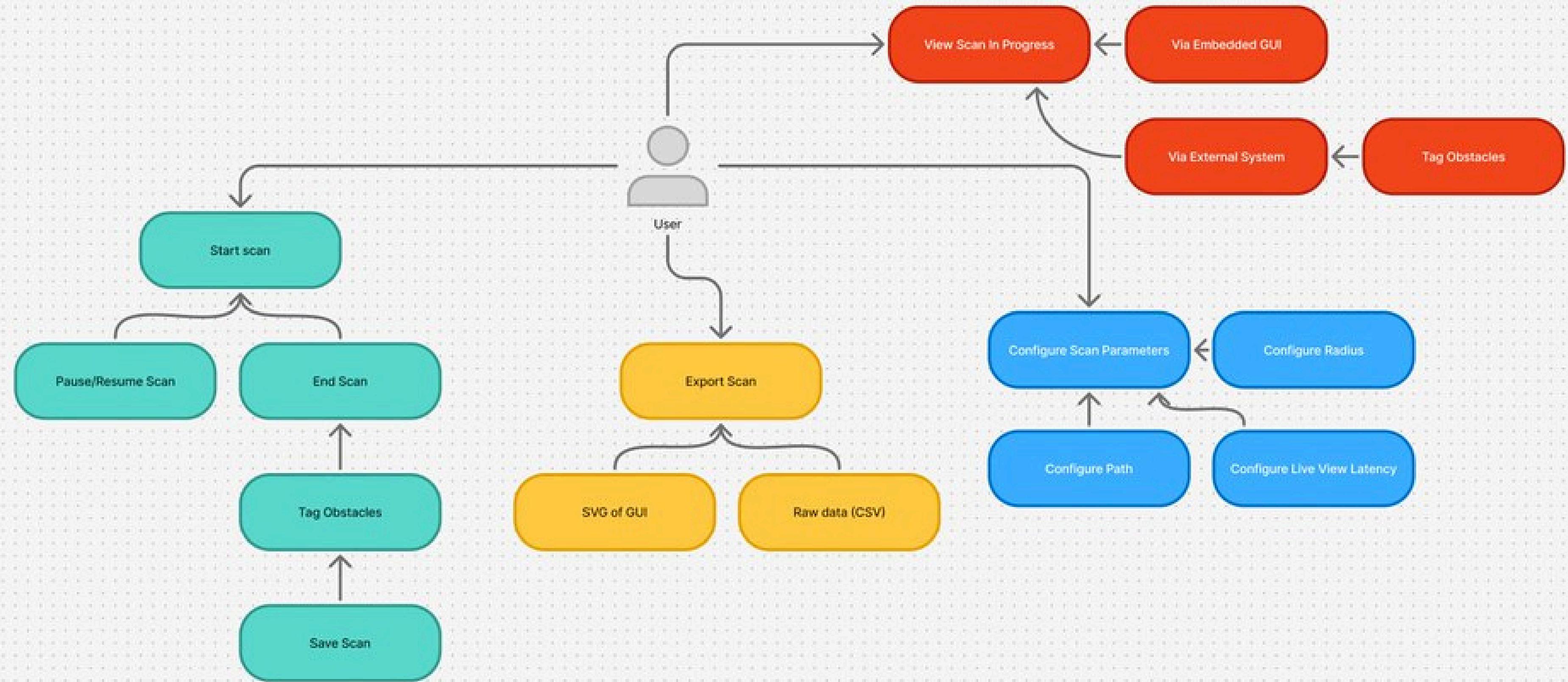
**Purpose:** Illustrates the interactions between the user and the key system functionalities, highlighting the user's control over the scanning workflow, configuration, and data export.

## Key Features Represented:

- Start and Control Scan: Users can initiate, pause, and stop the LiDAR scan manually.
- Real-Time Map Visualization: Live 2D map view updates during scanning via GUI or external systems.
- Tagging and Saving Obstacles: Users can mark obstacles during or after the scan and save the session.
- Configure Scan Parameters: Radius, path, and latency can be adjusted before starting the scan.
- Export Data (CSV/SVG): Scanned map data can be exported as raw CSV or visual SVG format.

# USE CASE DIAGRAM

**Relation to Other Diagrams:** Complements the **Context Diagram** by showing user-level interaction rather than external system flow.



# SEQUENCE DIAGRAM

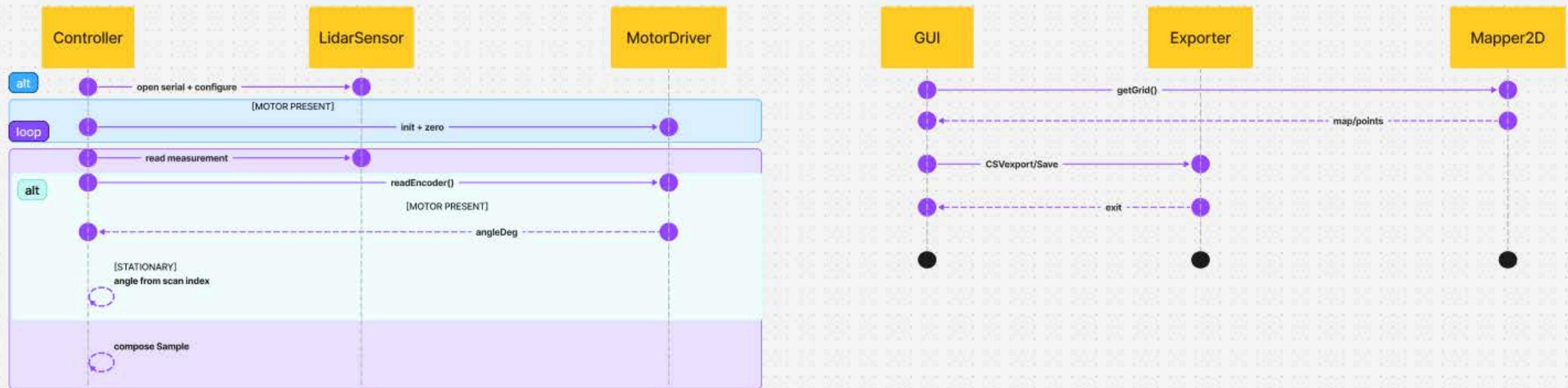
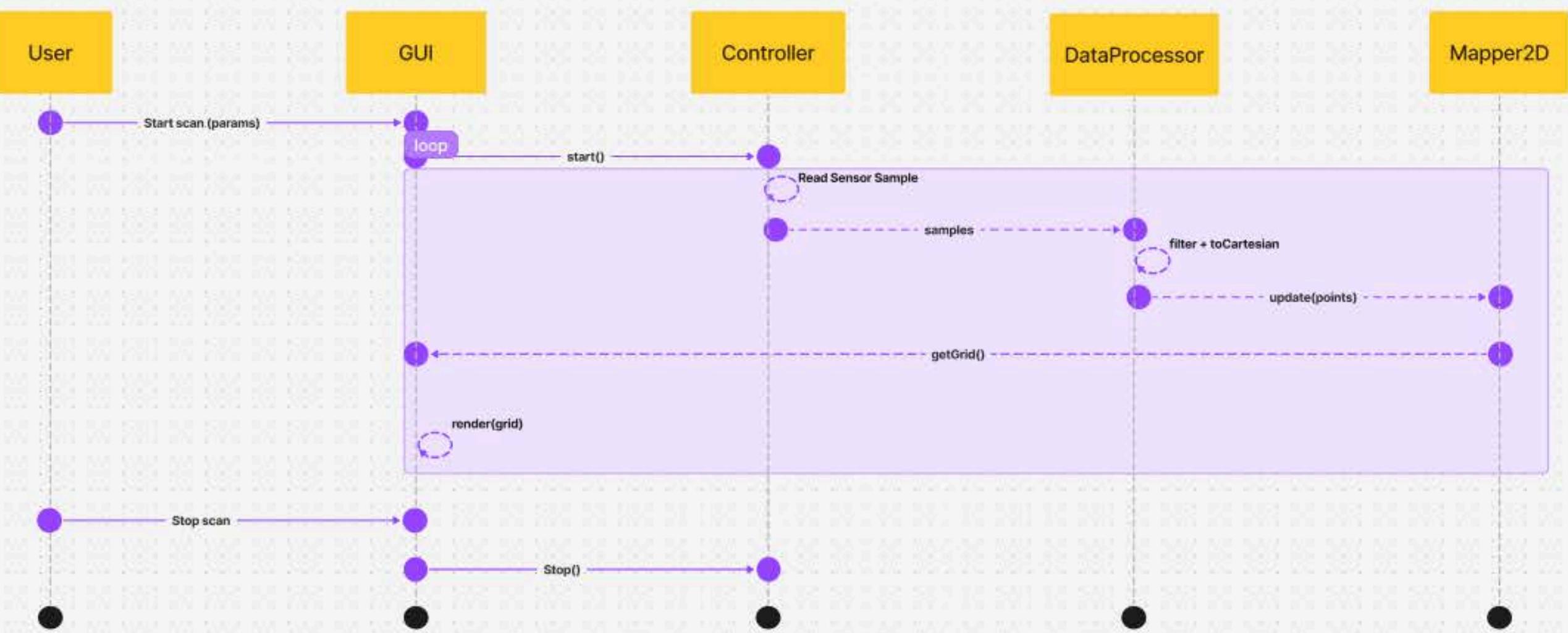
**Purpose:** Details the step-by-step interaction between system components during scanning, data processing, and exporting, showing the temporal order of operations.

## Key Features Represented:

- User-GUI Interaction to Start Scan: Scanning begins when the user inputs parameters through the GUI.
- Sensor and Motor Coordination: Controller synchronizes LiDAR readings with motor angle data.
- Data Filtering and Conversion: Raw sensor samples are processed into usable map coordinates.
- Dynamic Grid Update and Rendering: The 2D map is continuously updated and rendered in real time.
- Scan Data Export Workflow: GUI triggers export, and data is saved through the Exporter module.

# SEQUENCE DIAGRAM

**Relation to Other Diagrams:** Each sequence aligns with user actions from the use case diagram and reflects data flows shown in Data Flow Diagram/context diagrams.



# CLASS DIAGRAM

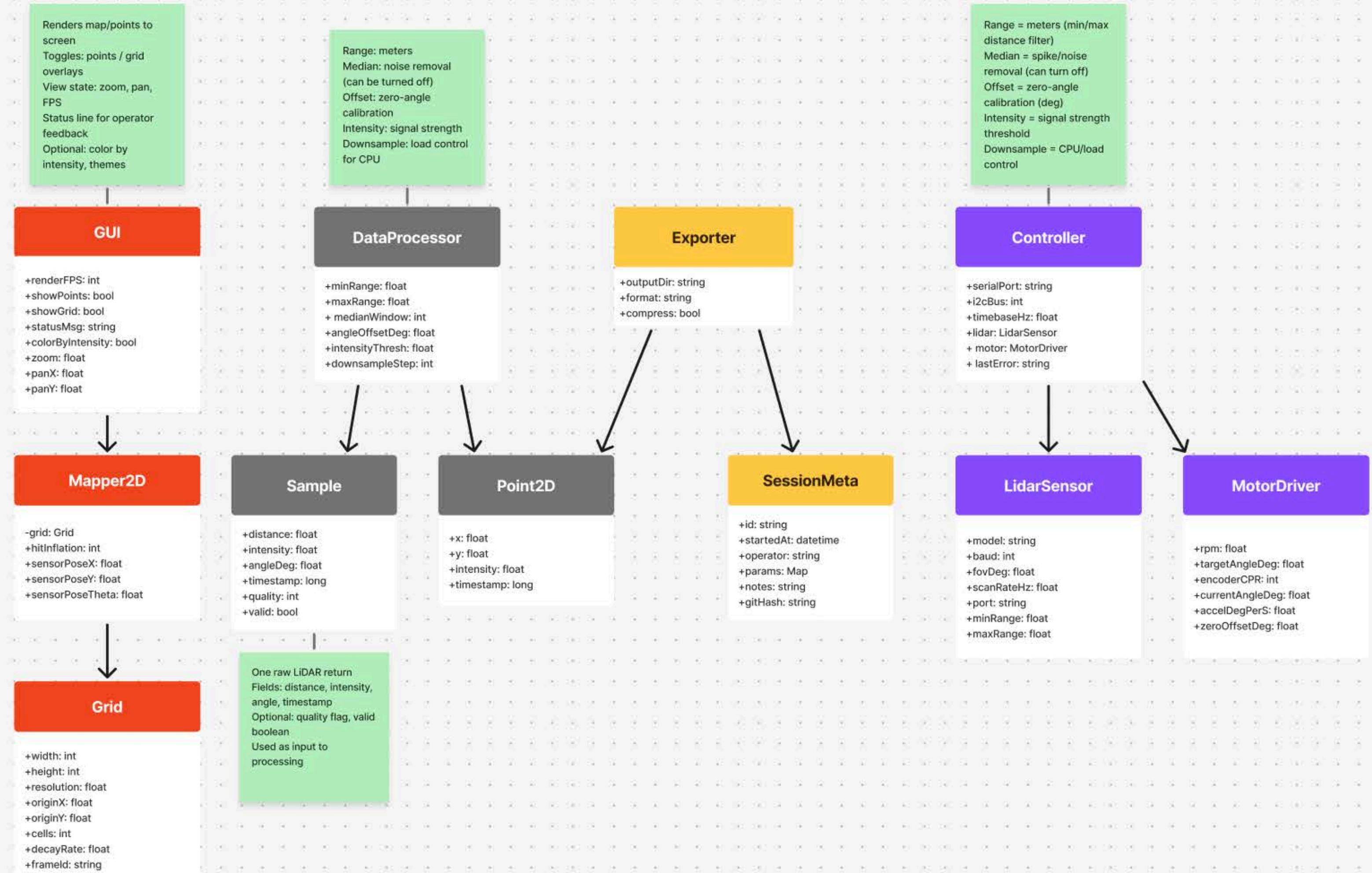
**Purpose:** Shows the object-oriented structure of the LiStars software, the main classes, their attributes, and how they interact.

## Key Features Represented:

- LidarSensor and MotorDriver: Handle sensor data and motor control.
- Controller: Central coordinator that manages timing, communication, and data flow.
- DataProcessor: Filters and converts raw LiDAR data to usable map points.
- Mapper2D and Grid: Build and store the 2D representation of the environment.
- GUI: Visualizes the map and provides user feedback.
- Exporter and SessionMeta: Handle data export and session management.

# CLASS DIAGRAM

**Relation to Data Flow Diagram:** The Data Flow Diagram illustrates what data moves between system components, while the Class Diagram shows how those processes are implemented in code through the interaction of different classes.



# UI MOCKUPS

Shows what the user will see when using the application

# 2D LiDAR Dashboard

Start Stop

## Controls

Range (10 m)



Angle (180°)



Idle

Export SVG

Export CSV

## LiDAR Visualization

2D Map Display



## Data Log

Timestamp: 2025-10-12 16:32

Objects Detected: 5

Closest Object: 2.3 m

# 2D LiDAR Dashboard

Start

Stop

Export SVG

Export CSV

## Controls

Range: 10 m



Angle: 180°



Scanning

## Visualization

2D Map Display

## Data Log

Timestamp: 2025-10-12 16:32

Objects Detected: 5

Closest Object: 2.3 m