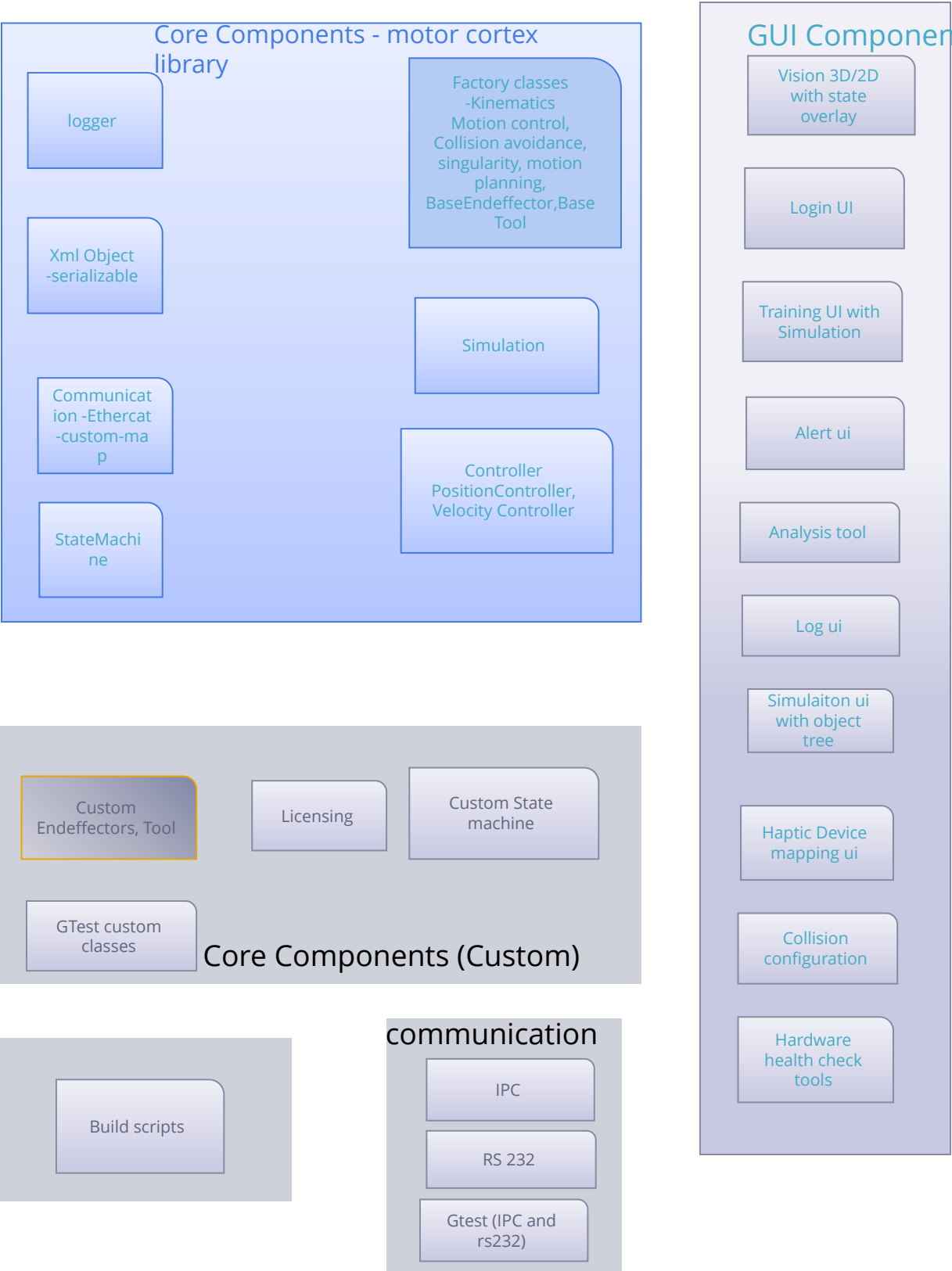


Software Architecture

Software component diagram



- Core Components

Core Components is a library that provides basic and common utilities for a robotic framework. It should provide factory classes for motion planning, end effector.

- General Utility Components
  - Xml Reader/Writer - C++ Object serialization/ deserialization library. It should provide the base class to read/ write all the data types like int, unsigned int, float, double, strings, buffer, maps, lists, vector, pair, images
  - Logger - It should support the ability to log all the basic data types like int, float strings with different flags/enum like warning, info, debug etc. It should allow the user to add own flags/enum. it can be turned; on and off during different software life cycle.
  - Simulation
- Factory classes
  - Kinematics - should deal with motion control, motion planning, collision avoidance mechanism, end effector support. It has to provide all base classes for end effector, motion planning etc
- Controller -
- State machine -basic default statemachine

- Custom Components,

- State machine implementation specifically for meril robot
- Custom motion movements implementation for meril robot
- Communication
  - IPC
  - RS 232

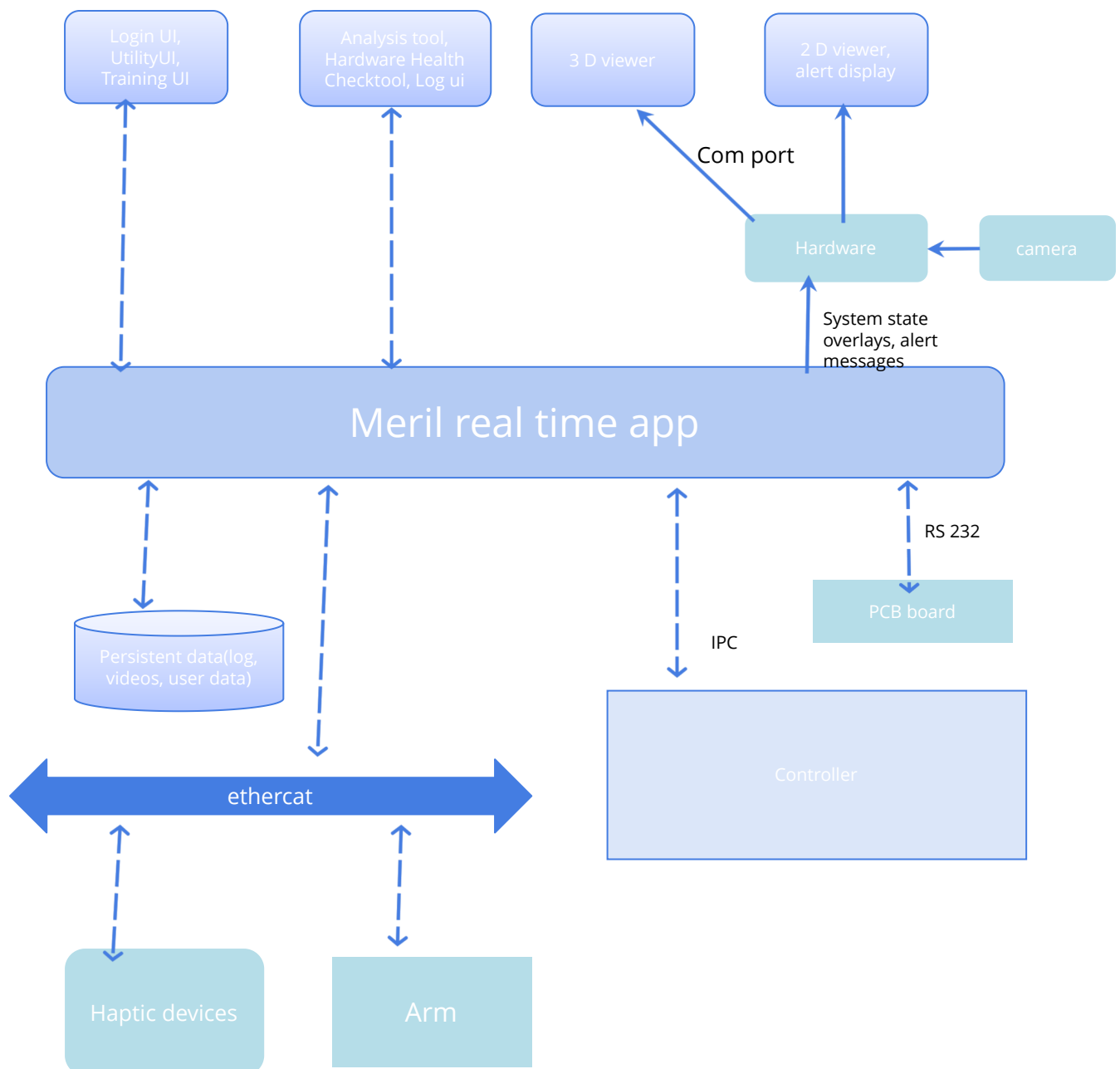
- GUI components

- UIs for surgeon or support staff
  - 3D/2D Viewer - with state machine , zoom facility
  - Login user
  - Training App - with simulation ui, with haptic devices, allows the surgeons to practice in simulated environment.
  - UI to set speed of haptic devices
- UIs for analysis, configuration, and debugging
  - Simulation UI - 3 D render ui, moves the simulation for dragger movements or haptic device

movements

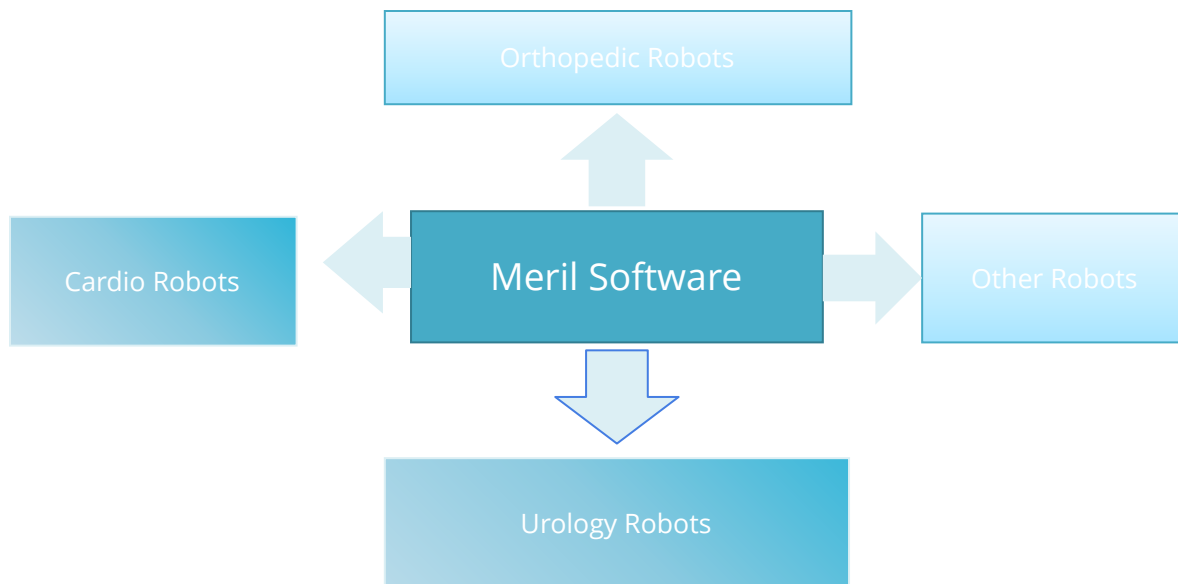
- Haptic Device mapping UI - UI to map each buttons or controls in haptic device to tools or manipulator link
- Log ui - It displays the useful information to the ui from the backend
- Alerts - quick alerts to get user attention specifically for collision or singularity
- Joint Configuration UI - to set joint limits, speed etc
- Analysis tools - to study the ability of surgeon or derive any data model out of it
- Collision configuration - ability to configure dynamic collision between different manipulators or collision between links of a manipulator
- Haptic device checker - ability to check the performance of the haptic device for example, for given movement - x of a particular device moves the arm in the same way always
- Hardware debugging Tools - to find the error state of the hardware

## User Interface



## Extensibility

The software should support all types of robots, whether it requires preplanned manipulation movements or dynamic movements as per haptic devices.



## Reusable

The software components should be modularised such that the components can be reusable.

## Process interaction Diagram

<TODO>

How many daemon are going to be there and where it is going to run?  
How many start up UIs?

## State Diagram