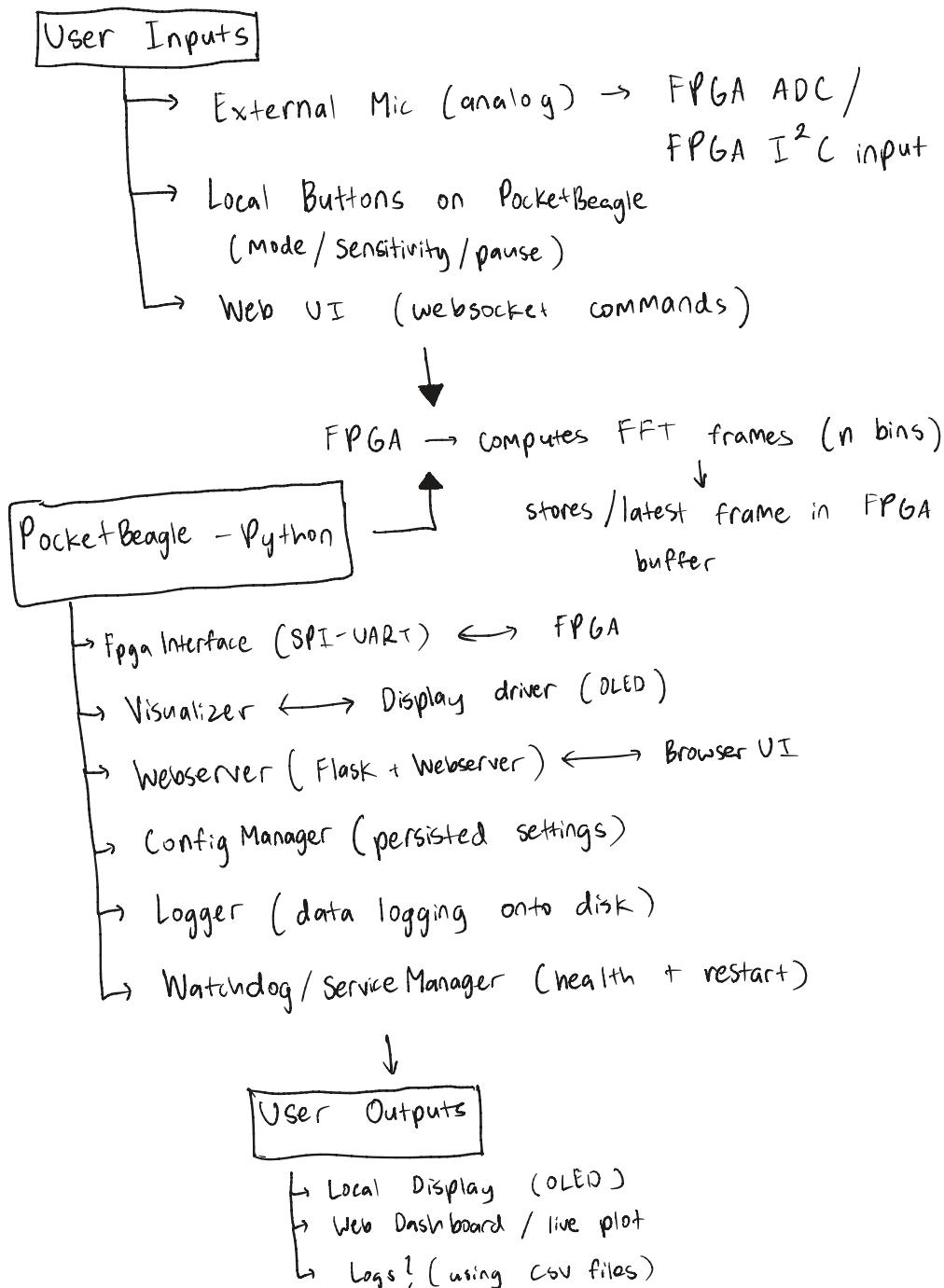


# Software Block Diagram



# Python Classes

## 1) Fpga Interface

L Manages communication w/ FPGA (reads FFT frames,  
Function: SPI / UART, CRC, framing, retries & parses into "FFT frame")  
Sends configuration

- Function: SPI / UART, CRC, framing, retries & parses into "FFT frame"

- Attributes: device, mode, freq-hz, frame-length

- Public methods:

- connect()

- read-frame() → returns FFTFrame or None

- send-config() → returns a boolean value

- close()

- Runs in its own thread

## 2) FFTFrame (data class)

- Fields: timestamp (float), bins (list [float, magnitude]), sample-rate, frame-id (int)

- Method: to\_dict() for logging / JSON

## 3) Visualizer

maps FFTFrame → pixels / LED intensities

- Function: scaling, smoothing, color mapping, peak hold, VU modes

- Attributes: display-driver (instance), scale-mode, smoothing-alpha

- Public methods:

- render-frame (frame: FFTFrame) → draws to display-driver

- set-mode (mode-name)

- autoscale-enable (on: bool)

- called from main loop of FPGA interface

Continued on next page ...

#### 4) Display Driver

Abstract base for hardware displays; implement concrete drivers:

- Function: hardware-specific drawing API (OLED over I<sup>2</sup>C,  
VGA via FPGA pins, NeoPixel via GPIO)

- Public Methods:

- init()
- draw\_pixels(buffer)
- clear()
- shutdown()

#### 5) WebServer (Flask + Websocket)

Provides remote UI + control

- Function: serve web dashboard, allow live updates, accept config changes

- Public methods / endpoints:

- GET / → HTML

- WS / live → JSON: FFTFrame.to\_dict()

- POST / config → change settings (FpgaInterface)

- subscribes to frames & broadcasts to connected clients

- runs in separate thread

#### 6) Config Manager

Persistent config

- Function: Load / save config

- Attributes: cfg dict, path

- Public methods:

- load(), save()

- get(key, default=None), set(key, val)

#### 7) Logger

Writes frame data & events to disk or CSV

- Function: time-series logging, rotation, debug logs

- Public methods: log\_frame(frame: FFTFrame), log\_event(msg: str)

## 8) Watchdog / Service Manager

- Monitors the System; ensures FpgalInterface alive; restarts subsystems
  - Function: health checks; triggers restart via systemd
- (fun fact: Systemd → system daemon)

## 9) Main App

Coordinates everything

- Public methods: run() (main loop), stop(), handle\_signal()