

# OBDII Architecture

*Vitaly Borodin, Zac Wisdom, Joseph Jenkins, Zac Slade*

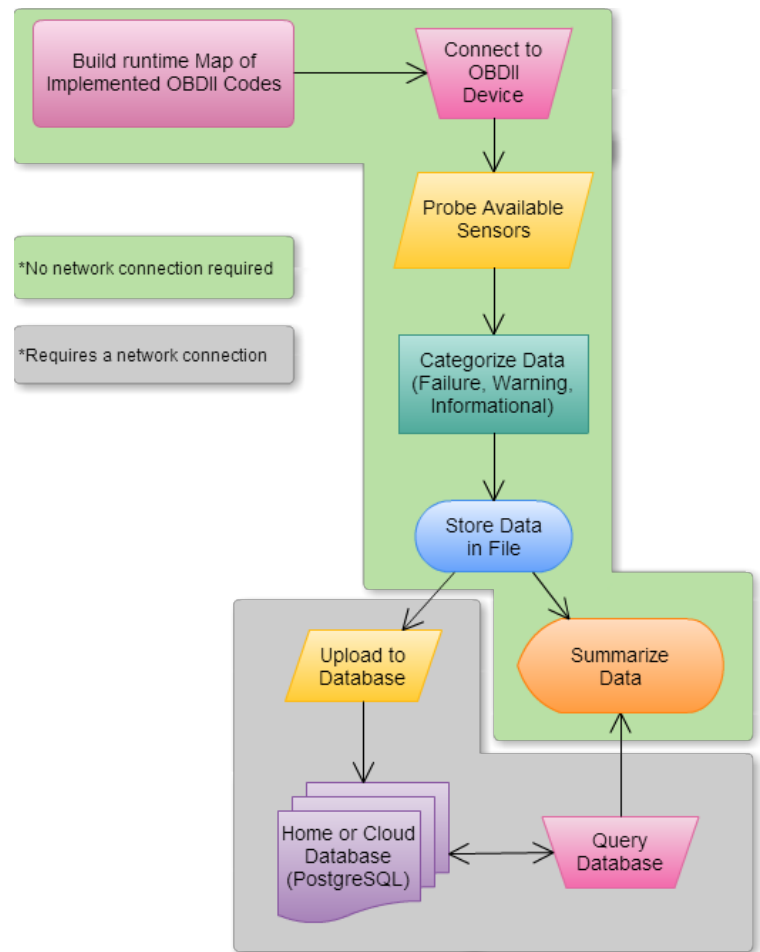
This program is to allow a user to monitor and collect information from a running car, or a car that has previously given some error warnings, whether this information is just live diagnostic data, or post error data. After collection, the information will be given in an understandable format to illustrate trends and possibly predict coming errors. The following diagram gives a high level breakdown of the program flow.

## OFFLINE

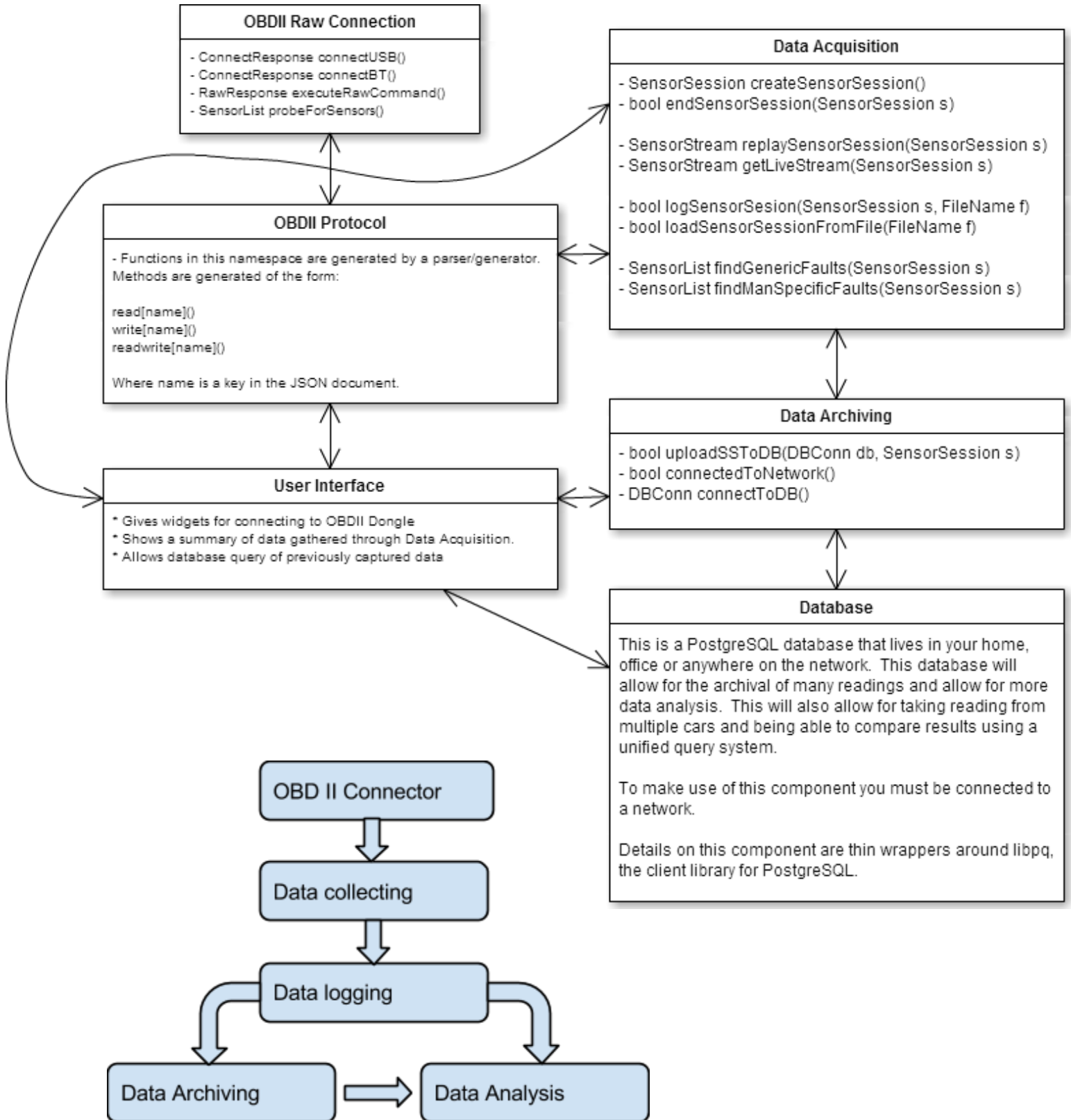
- Local database of generic and known manufacturer OBD-II codes to be read from the connected dongle. (Prior knowledge)
- OBD Dongle probes available car sensors to look up against database.
- Data is categorized into programmatic data structures in memory and on disk.
- This categorized data is then decoded for the end user.

## NETWORK CONNECTED

- Once connected, on disk data is uploaded to some centralized database (home, cloud, office, etc.)
- Data stored in centralized database can be queried by user for additional manipulations (predictive failures, total cost to own, projected MPG, etc.)

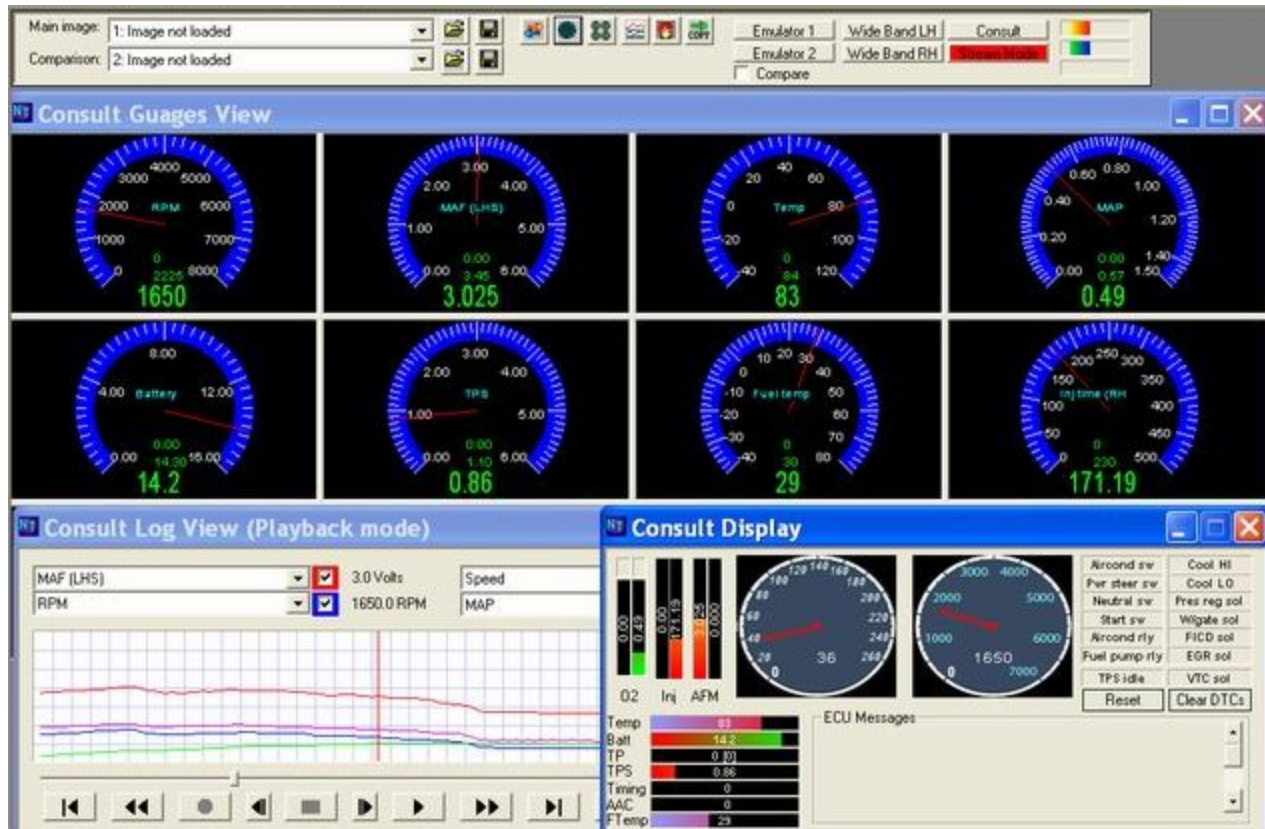


The following image(s) represent the component prototypes for the modules, and generalized methods for each to basically illustrate how the modules work internally.



## Sensors in a Typical Car

### Types of OBDII Data



## There are Several Ways to Interface with OBDII Devices

