Cootie Architecture

Node-red

Request → **Cootie Broker Node (e.g. Temp Sensor)** → Response

Cootie Broker Node (e.g. PWM)

Transport (Cootie Config Node)

Transports:
- Serial (USB)
- UDP
- HTTP
  - GET
  - POST
- Cootie
  - Client
  - Server

Request w/routing info

ALL messaging passed as JSON

Response w/routing info

"CootiePy" QTPy, Pico or other hardware

DEFINITION → **CootiePy, webserver, or other Endpoint**

Drivers

Interface Instances
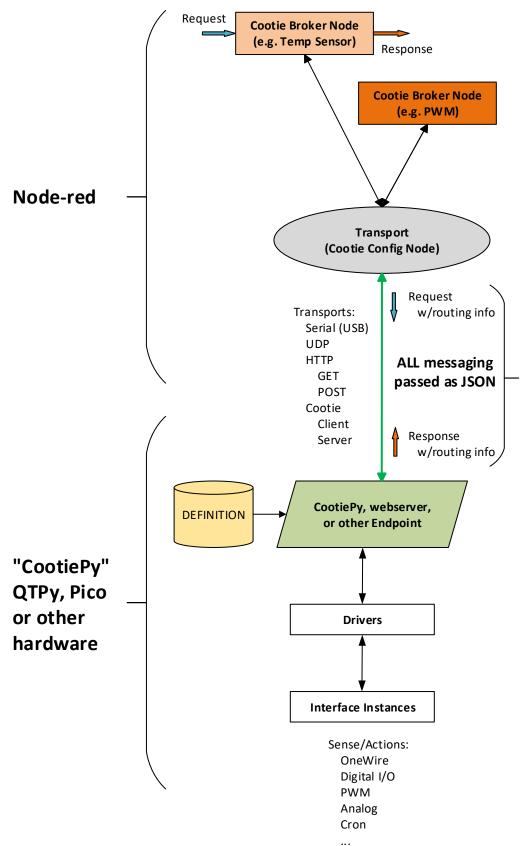
Sense/Actions:
- OneWire
- Digital I/O
- PWM
- Analog
- Cron
- ...

Broker nodes may be colored by groups, such as sensors, for better flow clarity.

Optional dedicated broker nodes may be defined for CMD, ACK, and ERR reponses as desired to aggregate messages.

Mulitple Cootie Broker nodes may connect to the same transport configuration.

Mulitple transports of the same type and/or different types may be configured for use.

Critical that the endpoint return routing info from the request in the response.

Endpoint can make multiple responses for a given request as long as routing info is maintained. For example, real time pin changes.

A single endpoint communicates with a specific transport.

A definition file matches "IDs" to interfaces and drivers abstracting messaging.

Drivers make for generic message interfaces to different hardware instances, e.g. all temperature sensors input and output same data regardless of type. Endpoint need only incorporate drivers needed for application.

With the information provided, the endpoint performs the specific actions and generates one or more responses.