# *gCoAP:* GNRC CoAP for RIOT

## **Why gCoAP**?

- ***RIOT-specific***
  - messaging, event loop, GNRC stack
  - Tailored to the RIOT community, LGPL
- ***Friendly***: well documented and accessible
- ***Goldilocks***: Just Right between complexity of libcoap and simplicity of microcoap

# Event Loop, Messaging

```
gnrc_coap_send(gnrc_coap_sender_t*, …) {
    xtimer_set_msg(timer, timeout, msg, coap_pid);
    gnrc_netapi_dispatch_send(...)
}

static void *_event_loop(void *arg) {
        case GNRC_NETAPI_MSG_TYPE_RCV:
                listener = gnrc_coap_listener_find(port);
                _receive(pkt, listener, src_addr, port);

        case GNRC_COAP_MSG_TYPE_TIMEOUT:
                _expire_request(gnrc_coap_sender_t*);
}
```

# Structs

- ***gnrc_coap_meta_t***: message ID, token
- ***gnrc_coap_transfer_t*** : path and payload (I/O representation, not resource itself)

- ***gnrc_coap_listener_t***: listens on a port for request or response
- ***gnrc_coap_sender_t***: sends a message, maintains message state and timeout, uses a listener to retrieve response