OPEN CONNECTIVITY FOUNDATION™

OCF on small devices using Soletta

Otavio Pontes - OTC - Intel

IoTivity

Soletta Overview

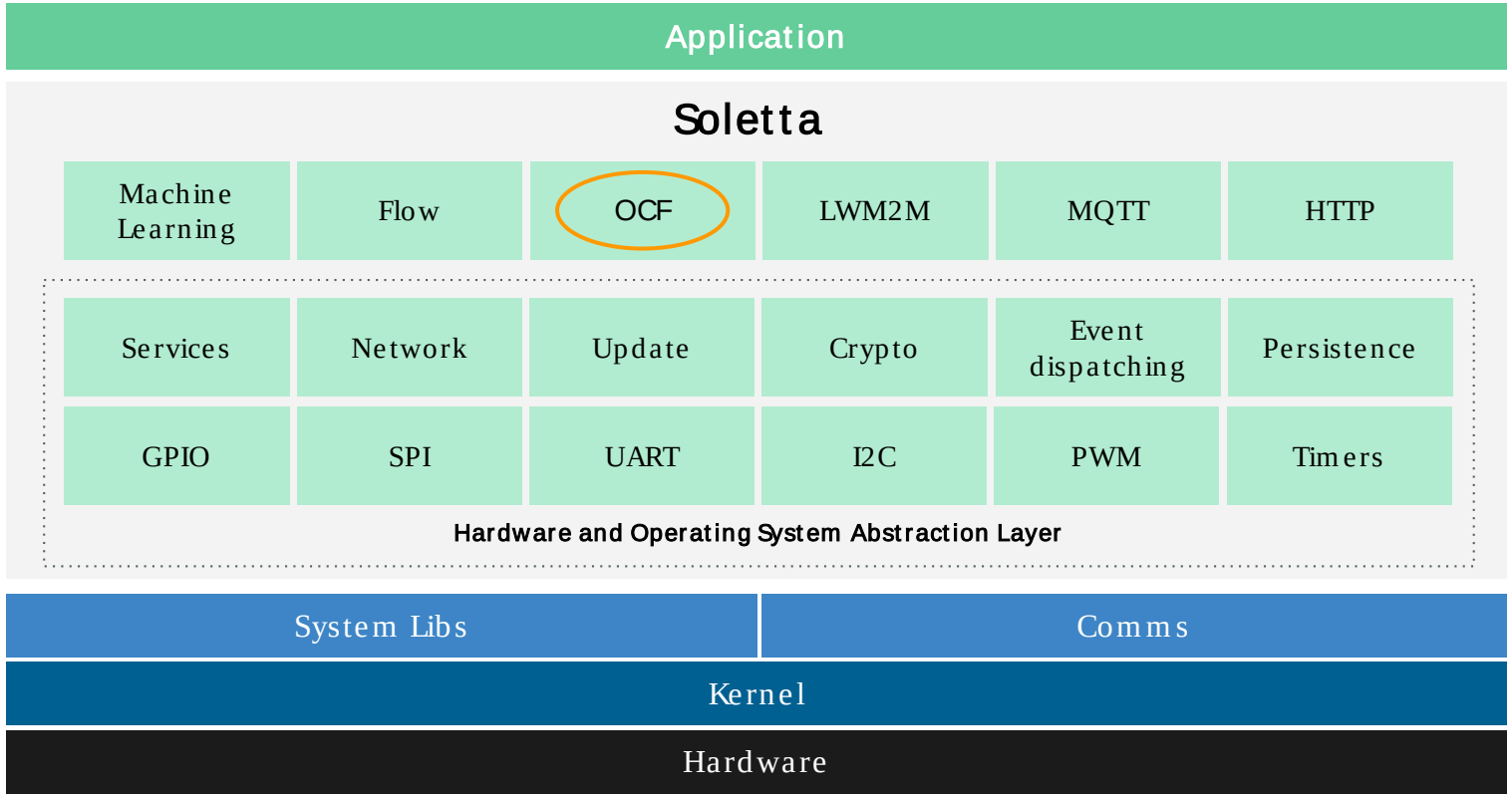
OPEN CONNECTIVITY FOUNDATION™

- IoT framework
- Open Source
- Easy access
  - Sensors
  - Actuators
  - Communication
- Portable code
- Different platforms, including small OSs

Soletta™ Project

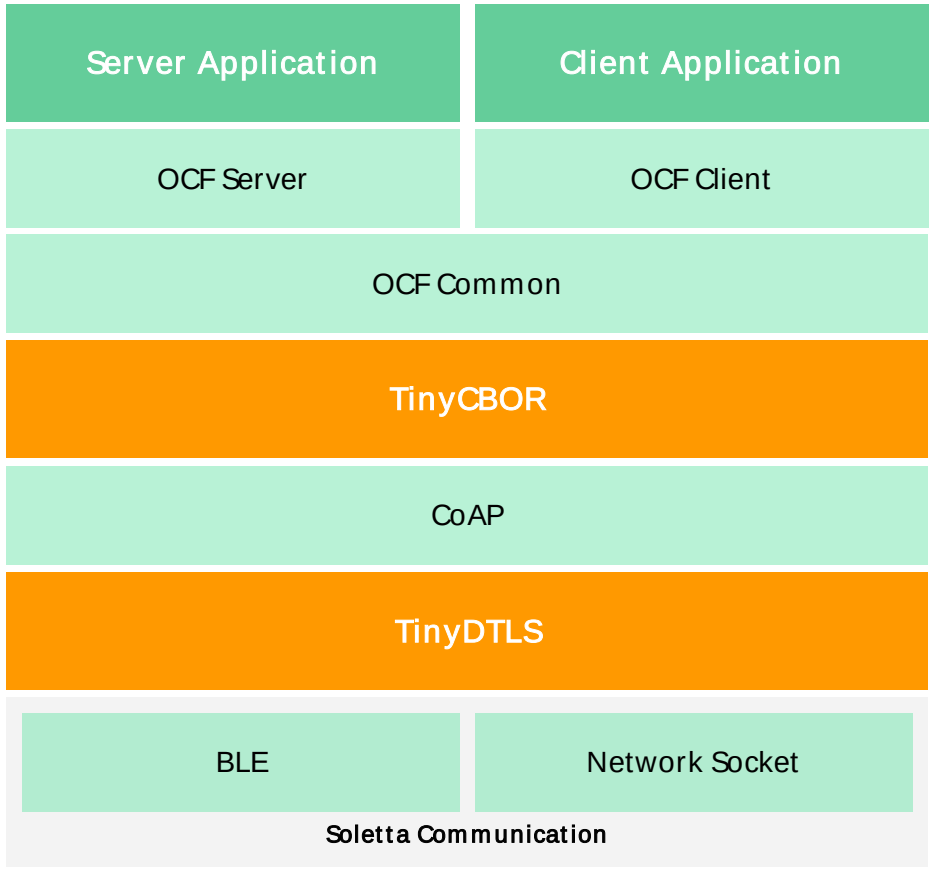
Soletta Overview

OPEN CONNECTIVITY FOUNDATION™



Soletta OCF implementation

OPEN CONNECTIVITY FOUNDATION™



Simple Light Sample

OPEN CONNECTIVITY FOUNDATION™

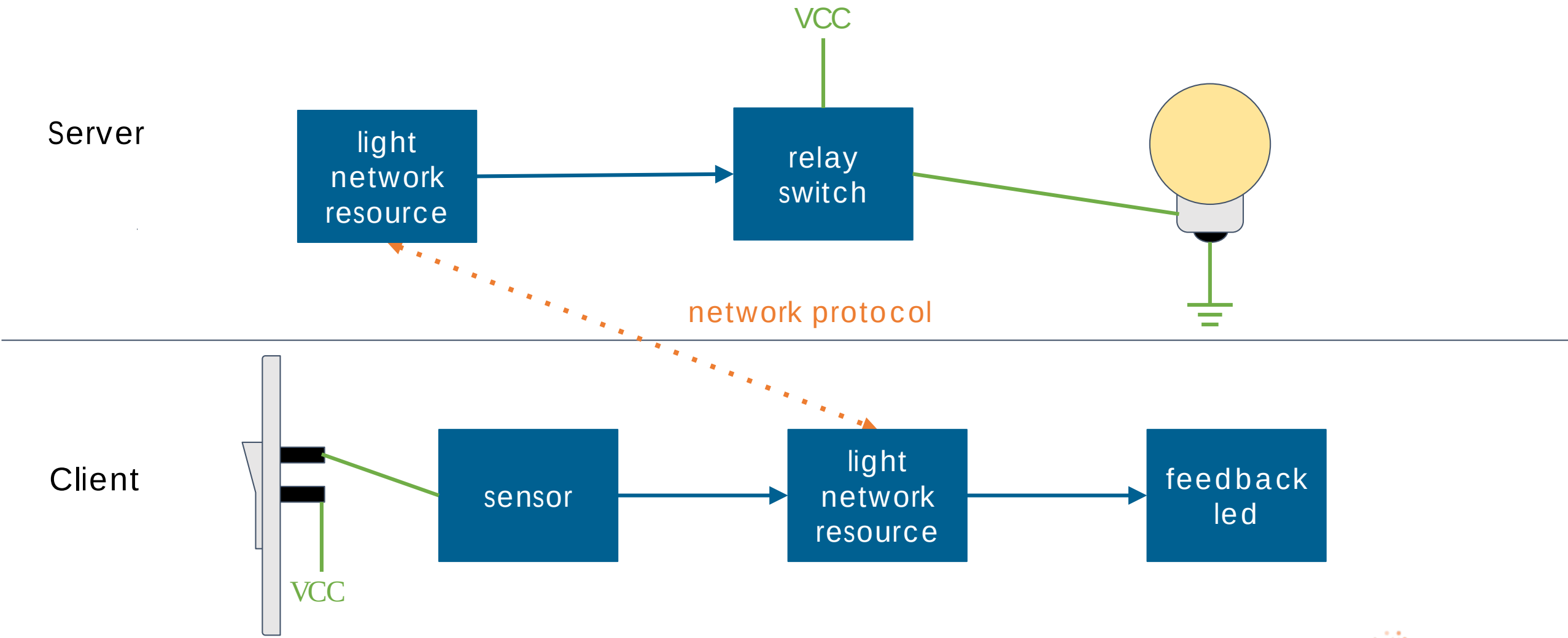
Simple and canonical example:

how to remotely toggle a light bulb?

IoTivity

Simple Light Sample

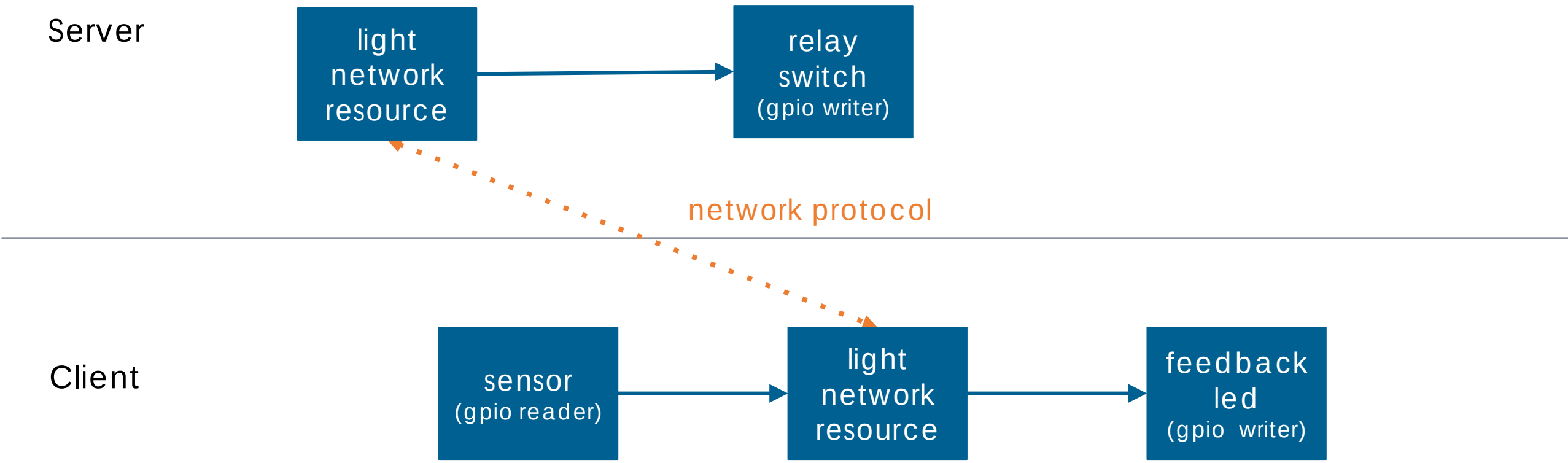
OPEN CONNECTIVITY FOUNDATION™



IoTivity

Simple Light Sample (Software)

OPEN CONNECTIVITY FOUNDATION™



IoTivity

Sample Light- FBP

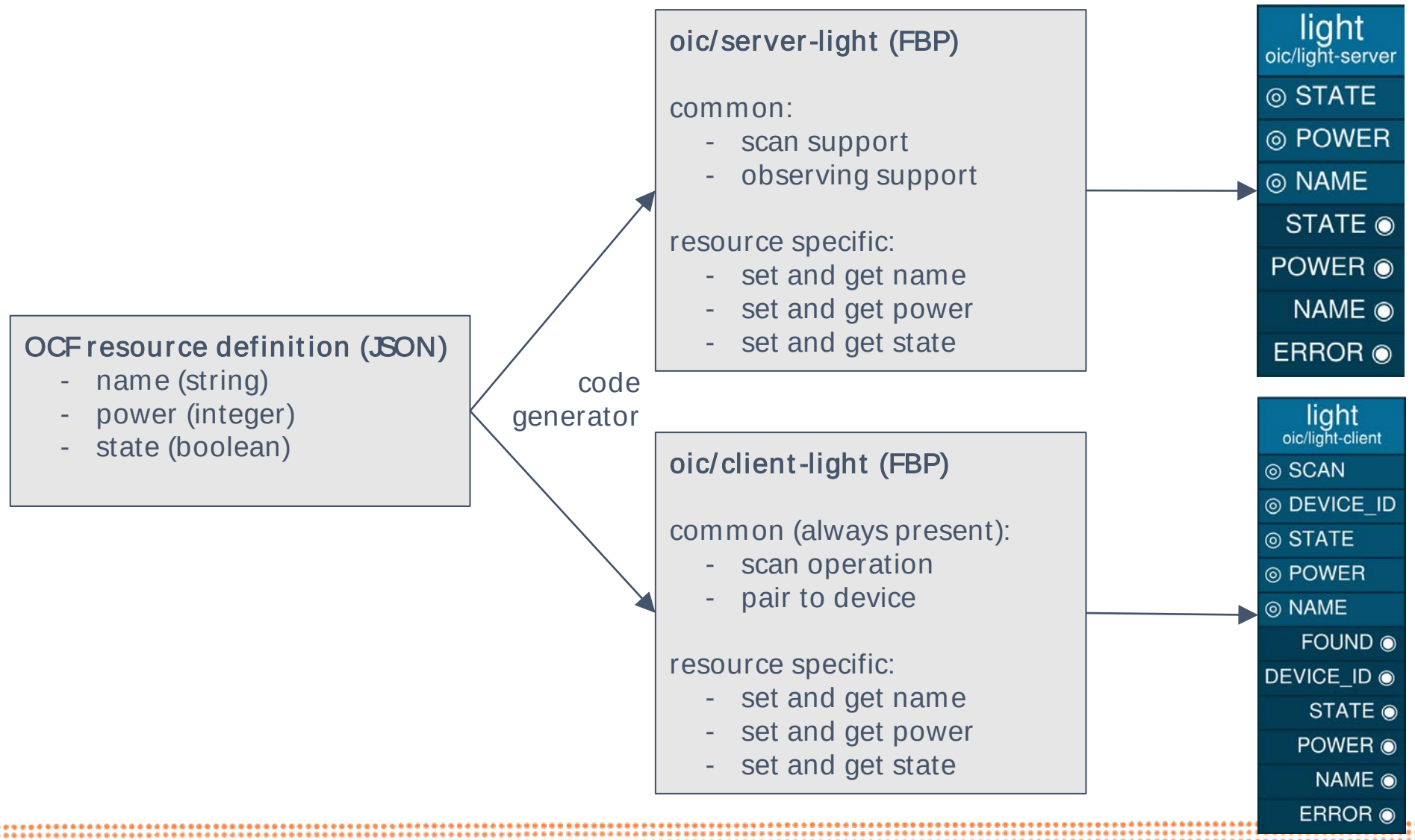
OPEN CONNECTIVITY FOUNDATION™

```
#Server
light(oic/server-light)
led(gpio/writer:pin=3)

light STATE -> IN led
```


```
#Client
button(gpio/reader:pin=7)
#Update device_id with server device id
light(oic/client-light:device_id="")
feedback_led(gpio/writer:pin=3)

button  OUT -> STATE light
light STATE -> IN feedback_led
```

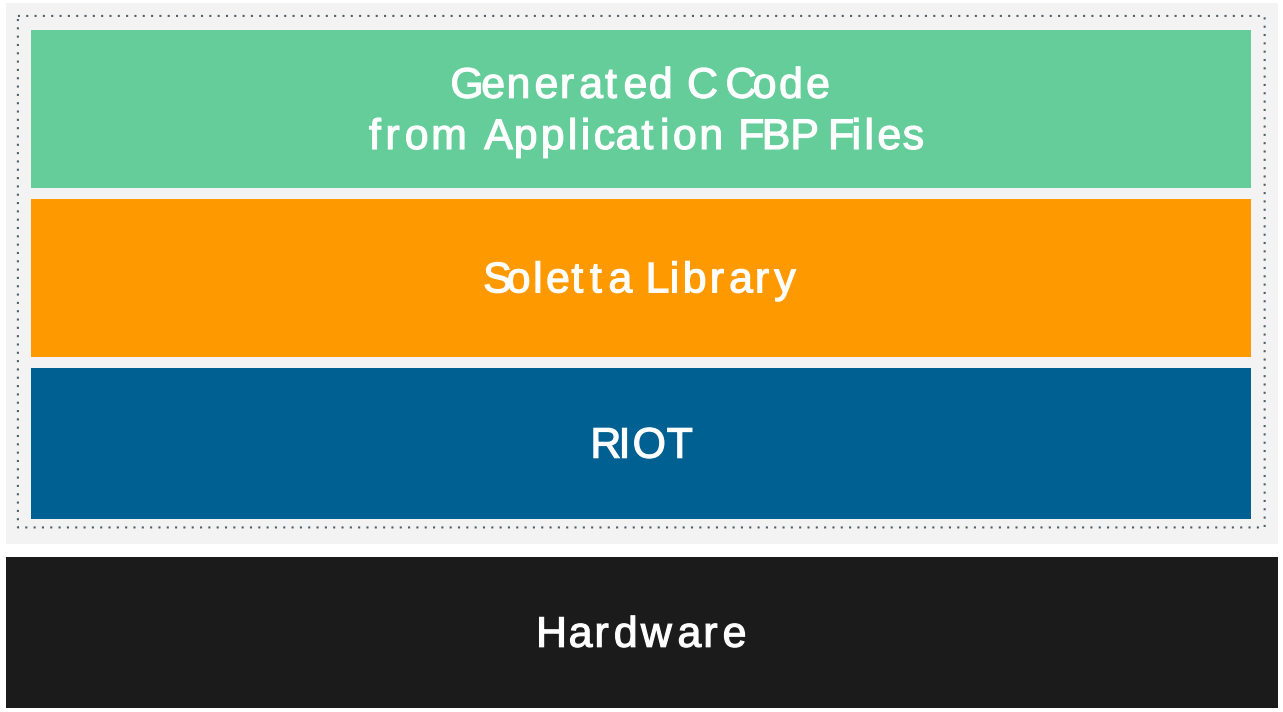
IoTivity

OCF nodes


OCF resource definition (JSON)
- name (string)
- power (integer)
- state (boolean)
code generator
oic/server-light (FBP)
common:
- scan support
- observing support
resource specific:
- set and get name
- set and get power
- set and get state
oic/client-light (FBP)
common (always present):
- scan operation
- pair to device
resource specific:
- set and get name
- set and get power
- set and get state
light
oic/light-server
STATE
POWER
NAME
STATE
POWER
NAME
ERROR
light
oic/light-client
SCAN
DEVICE_ID
STATE
POWER
NAME
FOUND
DEVICE_ID
STATE
POWER
NAME
ERROR

Small OS: Measurements

OPEN CONNECTIVITY FOUNDATION™



- RIOT
  - Image size:192k
  - Peak used RAM: around 10k

IoTivity

Sample server using C API

OPEN CONNECTIVITY FOUNDATION™

```
//Showing only code highlights and omitting error handling
int main(int argc, char *argv[]) {
    //[..] variables and non-soletta code
    sol_init(); //Initiate soletta

    res = sol_oic_server_add_resource( //Add resource to OIC server
        &my_resource, //struct used to describe the resource
        user_data,   //User's data to be passed to resource callbacks
        SOL_OIC_FLAG_DISCOVERABLE //This resource is discoverable
        | SOL_OIC_FLAG_OBSERVABLE //This resource is observable
        | SOL_OIC_FLAG_ACTIVE);   //This resource is active

    sol_run(); //Start soletta Main Loop. Will block until sol_quit()

       sol_shutdown(); //Shutdown soletta and clean up resources
}
```

IoTivity

Sample server using C API

OPEN CONNECTIVITY FOUNDATION™

```
struct sol_oic_resource_type my_resource = {
    .resource_type = sol_str_slice_from_str("core.led"),
    .interface = SOL_STR_SLICE_LITERAL("oc.mi.def"),
    .get.handle = my_handle_get,   //User callback, called with user_data
    .put.handle = my_handle_put    //User callback, called with user_data
};
```

IoTivity

Sample server using C API

OPEN CONNECTIVITY FOUNDATION™

```
static sol_coap_responsecode_t
my_handle_get(const struct sol_network_link_addr *cliaddr,
              const void *user_data, //What was given in server_add_resource
              const struct sol_oic_map_reader *input, //CBOR reader of payload
              struct sol_oic_map_writer *output) //CBOR writer of response
{
    struct sol_oic_repr_field field;

    field = SOL_OIC_REPR_BOOLEAN("state", //Resource property as defined by OCF
                                 getLedState(user_data)); //Application's code
    sol_oic_map_append(output, &field);

    return SOL_COAP_RSPCODE_CONTENT;
}
```

IoTivity

Sample server using C API

OPEN CONNECTIVITY FOUNDATION™

```
static sol_coap_responsecode_t
my_handle_put(const struct sol_network_link_addr *cliaddr,
              const void *user_data, //What was given in server_add_resource
              const struct sol_oic_map_reader *input, //CBOR reader of payload
              struct sol_oic_map_writer *output) //CBOR writer of response
{
    enum sol_oic_map_loop_reason reason; //Used by CBOR loop helper to inform why it ended
    struct sol_oic_repr_field field; //Used by CBOR loop helper to store the current field
    struct sol_oic_map_reader iter; //Used by CBOR loop helper to keep state

    SOL_OIC_MAP_LOOP(input, &field, &iter, reason) { //CBOR loop helper
        if (!strcmp(field.key, "state") && //Resource property as defined by OCF
            field.type == SOL_OIC_REPR_TYPE_BOOLEAN) {
            if (setLed(user_data, field.v_boolean)) //Application's code
                return SOL_COAP_RSPCODE_OK;
            return SOL_COAP_RSPCODE_INTERNAL_ERROR;
        }
     }
     return SOL_COAP_RSPCODE_BAD_REQUEST;
}
```

IoTivity

Sample client using C API

OPEN CONNECTIVITY FOUNDATION™

```
struct Context { //User's data
    struct sol_oic_client *client; //
    struct sol_oic_resource *res;
};
int main(int argc, char *argv[]) {
    //[...]
    struct Context ctx = { 0 }; //Create user's data struct

    sol_init(); //Initiate soletta

    ctx.client = sol_oic_client_new(); //Create a new client
    //Start discovery, looking for a core.led
    sol_oic_client_find_resource(client, &srv_addr, "core.led", //Resource type
            found_resource_cb, &ctx); //Callback and callback data

    sol_run(); //Start soletta Main Loop
    sol_oic_client_del(ctx.client); //Clean up
    return 0;
}
```

IoTivity

Sample client using C API

OPEN CONNECTIVITY FOUNDATION™

```
static bool
found_resource(struct sol_oic_client *cli, struct sol_oic_resource *res, void *data)
{
    struct Context *ctx = data; //User's data

    if (!res) { //Discovery timeout
            printf("Discovery timeout\n");
            return false;
    }

    //User's implementation: Check if this is the needed resource

    printf("Observing resource: %.*s\n", SOL_STR_SLICE_PRINT(res->href));
    ctx->res = sol_oic_resource_ref(res); //Keep discovered resource.
    //Start observing resource
    sol_oic_client_resource_set_observable(cli, res, resource_changed_cb, ctx, true);
        return false; //false means we can stop discovery process
}
```


IoTivity

Sample client using C API

OPEN CONNECTIVITY FOUNDATION™

```
static void
resource_changed_cb(sol_coap_responsecode_t response_code, struct sol_oic_client *cli, const
struct sol_network_link_addr *cliaddr,
    const struct sol_oic_map_reader *input, void *users_data)
{
    enum sol_oic_map_loop_reason reason; //Used by CBOR loop helper to inform why it ended
    struct sol_oic_repr_field field; //Used by CBOR loop helper to store the current field
    struct sol_oic_map_reader iter; //Used by CBOR loop helper to keep state

    SOL_OIC_MAP_LOOP(input, &field, &iter, reason) { //CBOR loop helper
            if (!st     rcmp(field.key, "state") && //Resource property as defined by OCF
            field.type == SOL_OIC_REPR_TYPE_BOOLEAN) {
            setFeedbackLed(users_data, field.v_boolean);  //Application's code
            return;
        }
    }
}
```

IoTivity

Sample client using C API

OPEN CONNECTIVITY FOUNDATION™

```
//Application's code
static void
button_callback(Context *ctx)
{
    struct Context *ctx = data;
        static bool state = false;

     //Check if we have discovered the resource
        if (!ctx->res)
            return true;

        state = !state; //Change button's state
            sol_oic_client_resource_request(
            ctx->client, ctx->res,
            SOL_COAP_METHOD_PUT, //coap method
            fill_packet_data, &state,
            NULL, NULL); //Request callback

        return true;
}
```

```
//Callback to fill packet data
static bool
fill_packet_data(void *data, struct
sol_oic_map_writer *repr_map)
{
        bool *state = data;

    //Helper to write CBOR content
        sol_oic_map_append(repr_map,
            &SOL_OIC_REPR_BOOLEAN("state"
    ,
            *state));
    return true;
}
```

IoTivity

What's next

OPEN CONNECTIVITY FOUNDATION™

- Improve memory consumption of observer's handling
- Automatic encode/decode from CBOR to structures in C
- Flow clients to support multiple observing servers
- Implement secure pairing process

IoTivity

Community

OPEN CONNECTIVITY FOUNDATION™

- IRC: #soletta @freenode
- Mailing Lists: https://lists.solettaproject.org/
- Code: https://github.com/solettaproject/
- Wiki: https://github.com/solettaproject/soletta/wiki
  - Site: https://solettaproject.org

IoTivity