Amarula Friday 30th of April 2026



Roadmap for newbies

Annachiara Gallo

www.amarulasolutions.com

Slides under Creative Commons license BY-SA 3.0.

1. Goal of this presentation

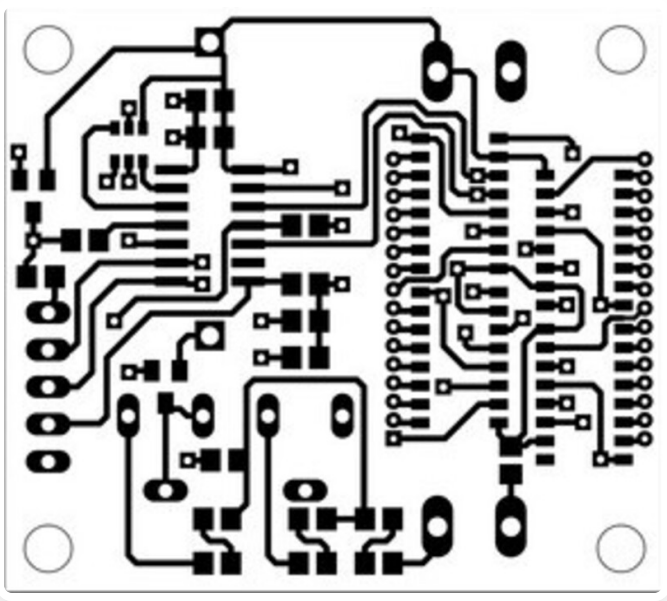
Case study: Task 139

- What goes into play when a junior is tasked with something they have little experience with?
- What blueprint can we lay out when such cases ensue?
- What are the most onerous steps of the process?
- Provide the client with this information

We can use this task to explore this questions, as with the task features themselves

2. Description of the task

Board



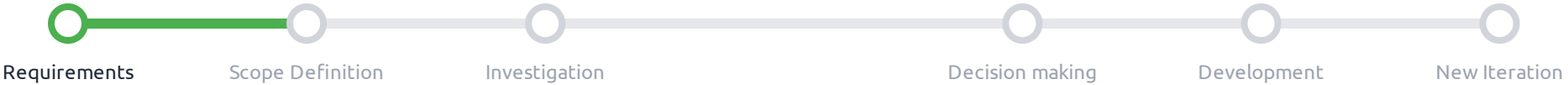
Starting Point (At commit `0ad02d0`)

- Wifi interface, AP and STA mode
- Https server (recent ssl migration)
- `mdns` responder to answer client's queries
- SSL self-signed certificates embedded, issued for hostname `https://board.local`, (browser warning)
- Hostname default 'board', modifiable via AT command
- Reachable via `https://board.local`

The New Requirement

The client wants to be able to reach the server with subdomain queries as well, for example `https://api.board.clientname.local`, `https://www.board.clientname.local`, and so on.

3. Iteration 1 - Expand requirements

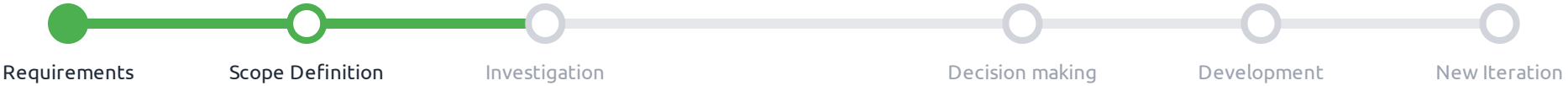


What are the requirements that were left implicit?

Are you familiar enough with the code and task context to be aware of them?

- Need of sequential hostname for each board
- Fallback hostname in case of failure
- Add command for reset of hostname
- Modification of certificate generation

4. Iteration 1 - Narrow the field of interest



Without a valid starting point, you open the gates for a flood of unnecessary information

Encircle yourself within a reasonably narrow perimeter of interest

Consider the assumptions you can make

- Client is not expected to have a proprietary domain
- HTTPS clients expected to run on Windows
- Hostnames variable, will have three labels plus the domain

Identify what information you actually need

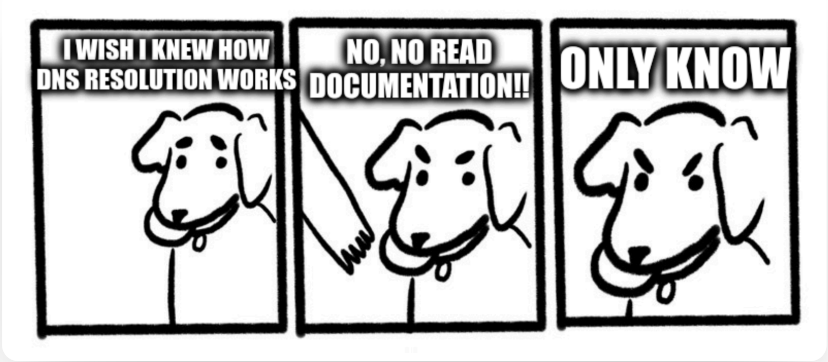
- How does Windows DNS resolution work?
- How does resolution of queries change with more than one label (one of which variable)?
- How does generation of certificates change with more than one label?
- What causes the browser to reject the current certificate?

5. Iteration 1 - Gather information and validate hands-on

Requirements | Scope Definition | Investigation | Decision making | Development | New Iteration

Organize the information you gain

- Windows DNS resolution handles `.local` queries in a non-standard way for historical reasons
- Multi-label queries might not be resolved by mDNS despite ending with `.local`
- The `.local` TLD is not recommended for anything other than mDNS queries
- Since 2015 SSL certificates must be signed by a CA for the browser to accept them
- Browsers will reject the certificate unless hostname is also listed inside the Subject Alternative Name (SAN) extension of the certificate.



6. Iteration 1 - Gather information and validate hands-on

Requirements | Scope Definition | Investigation | Decision making | Development | New Iteration

Test with single-label.internal

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.175.40.151	10.175.40.168	DNS	75	Standard query 0x49a7 A prova2.internal
2	0.000009	10.175.40.151	10.175.40.168	DNS	75	Standard query 0x49a7 A prova2.internal
3	0.000011	10.175.40.151	10.175.40.168	DNS	75	Standard query 0x49a7 A prova2.internal
4	0.000012	10.175.40.151	10.175.40.168	DNS	75	Standard query 0x49a7 A prova2.internal
5	0.000045	10.175.40.151	10.175.40.168	DNS	75	Standard query 0x49a7 A prova2.internal
6	0.000046	10.175.40.151	10.175.40.168	DNS	75	Standard query 0x49a7 A prova2.internal
19	0.301610	10.175.40.168	10.175.40.151	DNS	150	Standard query response 0x49a7 No such name A
20	0.301612	10.175.40.168	10.175.40.151	DNS	150	Standard query response 0x49a7 No such name A
21	0.301676	10.175.40.168	10.175.40.151	DNS	150	Standard query response 0x49a7 No such name A
22	0.301679	10.175.40.168	10.175.40.151	DNS	150	Standard query response 0x49a7 No such name A
23	0.301682	10.175.40.168	10.175.40.151	DNS	150	Standard query response 0x49a7 No such name A
24	0.301684	10.175.40.168	10.175.40.151	DNS	150	Standard query response 0x49a7 No such name A

7. Iteration 1 - Decide the course of action

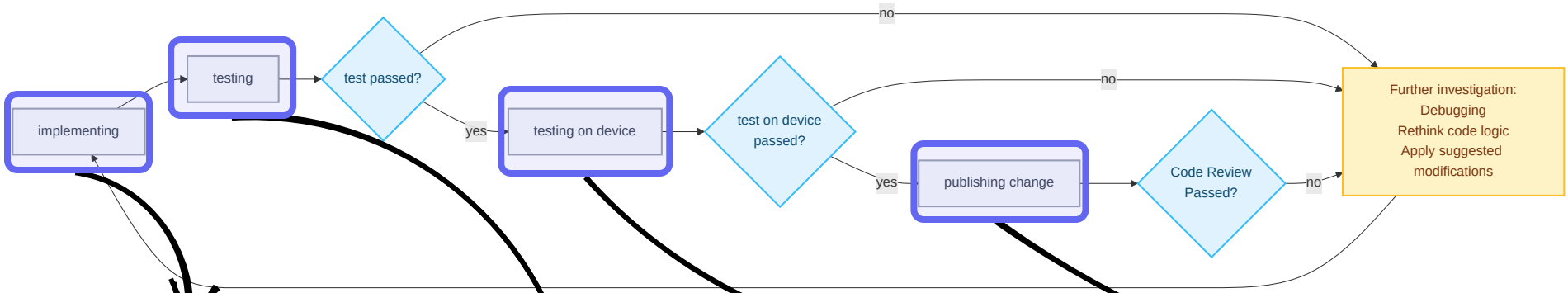
Requirements | Scope Definition | Investigation | Decision making | Development | New Iteration

- Inexperience is felt the most
- What should the fallback hostname be if the default setting fails?
- Get rid of `mdns` or propose a different approach (URL paths, dashes)?
- Try to evaluate yourself first, then detail the options to a senior



8. Iteration 1 - Development has its own iterations

Requirements | Scope Definition | Investigation | Decision making | Development | New Iteration



Not just coding

Focus on the files that will change

Plan for small, incremental commits

AT#HNAME= command

fallback name: using NIC

New certificates

Unit Testing

Validation of new variant of command

Validation of fallback name setting

Connection to server

Most time consuming part:

different initial states:

Verification in AP, STA mode

Verification with different clients

Push patch

Changes

Motivations

Related commits/tasks

9. Iteration 2 - Expand requirements

Requirements | Investigation | Decision Making | Development | End iteration

What are the requirements that were left implicit?

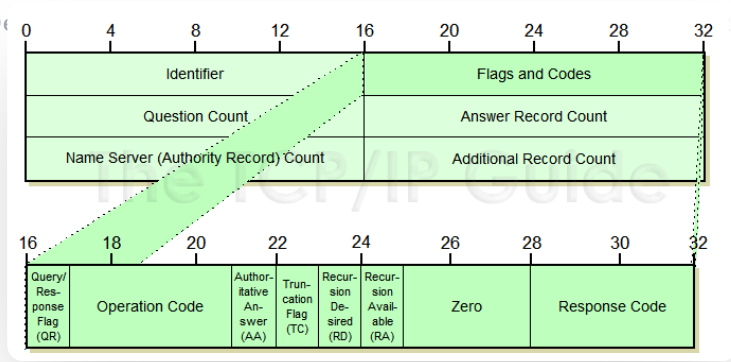
Are you familiar enough with the codebase to be aware of them?

- ~~Need of sequential hostname for each board~~
- Modification of certificate generation / set up CA to sign them
- Installation on Windows to be as user-friendly as possible
- Since `mdns` was removed, the board needs another way to let the client know it exists

10. Iteration 2 - Gather information and validate hands-on

Requirements — Investigation — Decision Making — De — eration

- Certificates: end of investigation (online guide for CA, `.bat` for installation)
- How does the device register its hostname on the network now that `mdns` is gone?



STA Mode

Code does not change but network configuration does
This must be tested by setting up a AP and a forwarder
Using `hostapd`, `dnsmasq`
Most important part to document since it must be explained to client

AP Mode

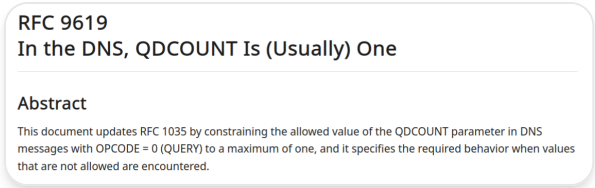
Implementation of a custom DNS responder
Need to familiarize with protocol, packet structure
Avoid common mistakes
Confront existing, reputable implementations: esp-idf `dns_server.c` example, which implements a captive portal, can be a starting point

11. Iteration 2 - Decision making

Requirements — Investigation — Decision Making — Development — End iteration

What to change and how?

- Each change could require a cascade of changes
- Code that had different use
- "They had a reason to write it like that"
- "There is no reason to keep that"
- **Example: updating the loop pointer**


RFC 9619
In the DNS, QDCOUNT Is (Usually) One

Abstract

This document updates RFC 1035 by constraining the allowed value of the QDCOUNT parameter in DNS messages with OPCODE = 0 (QUERY) to a maximum of one, and it specifies the required behavior when values that are not allowed are encountered.

esp-idf dns_server.c

```c
// Pointer to current answer and question
char *cur_ans_ptr = dns_reply + req_len;
char *cur_qd_ptr = dns_reply + sizeof(dns_header_t);
char name[128];

// Respond to all questions based on configured rules
for (int qd_i = 0; qd_i < qd_count; qd_i++) {
    char *name_end_ptr = parse_dns_name(cur_qd_ptr, name, sizeof(name));
    dns_question_t *question = (dns_question_t *)(name_end_ptr);

    /* .... */

    if (qd_type == QD_TYPE_A) {
        esp_ip4_addr_t ip = { .addr = IPADDR_ANY };
        // Check the configured rules to decide whether to answer this question or not
        for (int i = 0; i < h->num_of_entries; ++i) {
            // check if the name either corresponds to the entry
            if (strcmp(h->entry[i].name, "*") == 0 || strcmp(h->entry[i].name, name) == 0) {
```

12. Iteration 2 - Development: STA Mode

Requirements — Investigation — Decision Making — Development — End iteration

ap.conf

```
interface=wlp0s20f3
driver=nl80211
ssid=ESP_Test_Router

hw_mode=g
channel=6
wpa=2
wpa_passphrase=testpassword
```

dnsmasq.conf

```
interface=wlp0s20f3
bind-interfaces
dhcp-range=192.168.4.10,192.168.4.100,12h

dhcp-host=EC:DA:3B:46:9F:FC,192.168.4.50

host-record=board-9FFC.clientname.internal,192.168
address=/board-9FFC.clientname.internal/192.168.4.
```

hostapd | dnsmasq

NETWORK CONFIGURATION

- Set up AP and DNS resolver
- Assign IP
- Connect with other device
- Verify reachablity

13. Iteration 2 - Development: AP Mode

Requirements — Investigation — Decision Making — Development — End iteration

dns_server.c

```c
static const char *parse_dns_name(const char *req_start, const char *req_
                  char *parsed_name, size_t parsed_name_max_len)
{
    const char *current_ptr = name_ptr;
    const char *latch_after_jump = NULL;
    int already_parsed = 0;

    while (current_ptr < req_end) {
        uint8_t len_byte = (uint8_t)(*current_ptr);

        /* Name end reached */
        if (len_byte == 0) {
            if (latch_after_jump == NULL) {
                latch_after_jump = current_ptr + 1;
            }
            break;
        }

        /* First byte <after previous label> can be either length or compr
        if ((len_byte & DNS_PTR_FLAG) == DNS_PTR_FLAG) { /* Compr. ptr (>
            if (current_ptr + 1 > req_end) {        /* Next byte must be
```

```
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| 1  1|                 OFFSET                  |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
```

COMPRESSION POINTERS

DNS message compression scheme can be used to reduce the size of messages. When it is used, an entire domain name or several name labels are replaced with a (compression) pointer to a prior occurrence of the same name. The compression pointer is a combination of two octets: the two most significant bits are set to 1, and the remaining 14 bits are the OFFSET field.

```
          +----+----+----+----+----+----+----+----+----+----+----+----+
  +0x00 |    ID   |  FLAGS  | QDCOUNT | ANCOUNT | NSCOUNT | ARCOUNT |
          +----+----+----+----+----+----+----+----+----+----+----+----+
->+0x0c |0xc0|0x0c|   TYPE  |  CLASS  |0x04| t  | e  | s  | t  |0x03|
|         +----+--|+----+----+----+----+----+----+----+----+----+----+
| +0x18 | c  | o| | m  |0x00|  TYPE   |  CLASS  | ................  |
|         +----+--|+----+----+----+----+----+----+----+----+----+----+
|               |
-----------------
```

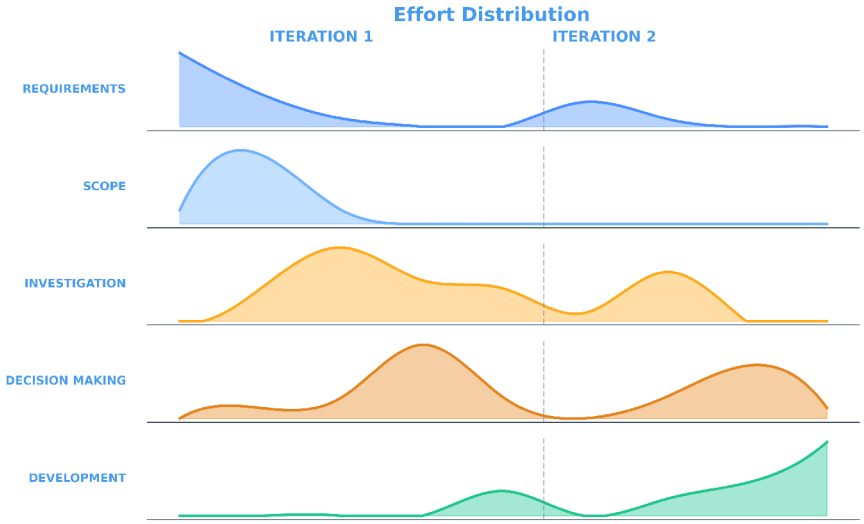
INFINITE LOOPS

The first occurrence of a compression pointer in an RR (an octet with the two highest bits set to 1) must resolve to an octet within a DNS record with

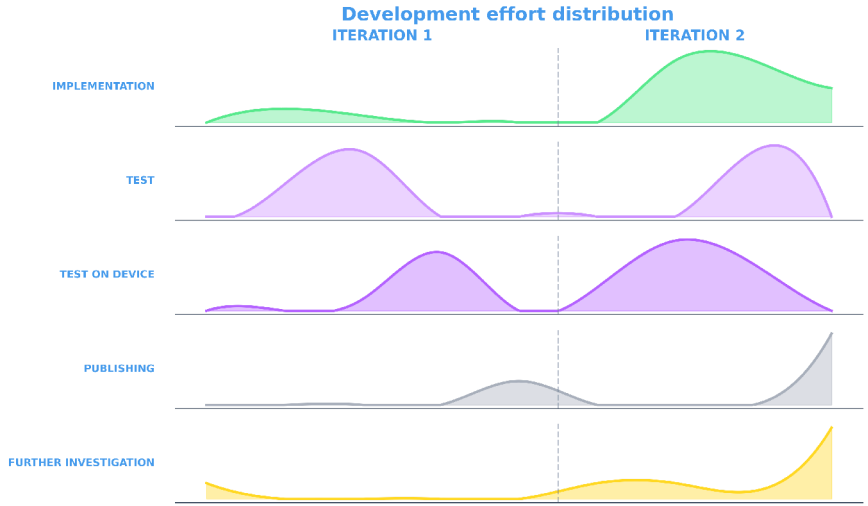
14. Final Considerations

What are the bottlenecks it terms of time and effort?

Main phases



Development breakdown



15. Documenting the task

- **Recording motivations**

  Explaining why specific choices were made prevents others from wasting time. In this specific case, it is important convey to the client the reasoning behind a network configuration requirment and an installation steup

- **Recycling spent effort**

  Especially for use cases that can be very common in IoT, such as solving the browser warning issue

- **Good for you as well**

  Will you remember all the logic behind your choices, or even just the testing steps 6 months from now? Or will you waste time second guessing?

> SPOILER: the second one



Visual representation of you 6 months from now

Thank you!

Thank You!

Any questions?