RFID
Reader / Cloner

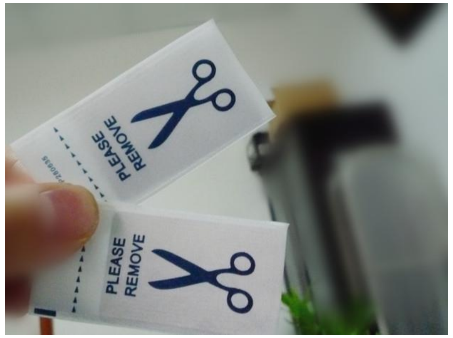
Author: **Iñaki Abadía Osta**

Tutor: **Enrique Torres Moreno**

Laboratorio de Sistemas Empotrados 2016/17 (LabSE)

Motivation

RFID



MIFARE

- **Classic**
- Ultralight
- DESFire
- Plus
- …

Motivation

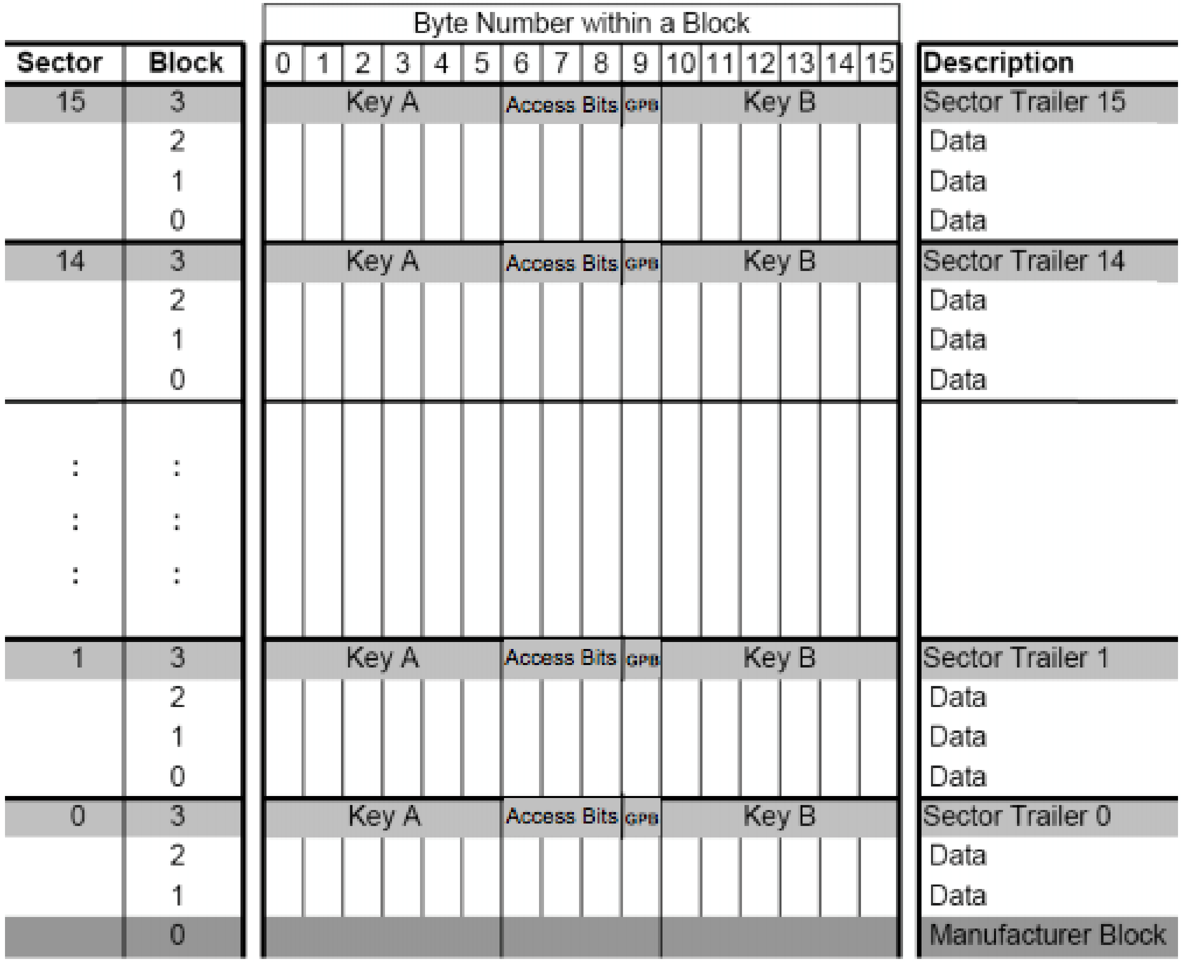
MIFARE CLASSIC

1KB (or 4)

Usually, only UID is used

Motivation

MIFARE CLASSIC 1K



		Byte Number within a Block																
Sector	**Block**	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	**Description**
15	3	Key A						Access Bits			GPB	Key B						Sector Trailer 15
	2																	Data
	1																	Data
	0																	Data
14	3	Key A						Access Bits			GPB	Key B						Sector Trailer 14
	2																	Data
	1																	Data
	0																	Data
:	:																	
:	:																	
:	:																	
1	3	Key A						Access Bits			GPB	Key B						Sector Trailer 1
	2																	Data
	1																	Data
	0																	Data
0	3	Key A						Access Bits			GPB	Key B						Sector Trailer 0
	2																	Data
	1																	Data
	0																	Manufacturer Block

- 64 Blocs, 16 Bytes
- 16 Sectors, 4 Blocs
  - 3 Usables
  - 1 Access bits, key

- 1024 Bytes
  - 768 usables

- Bloc 0
  - UID
  - **NO WRITABLE**

TEORETICALLY…

Motivation

MIFARE CLASSIC 1K = 0 SECURITY

Key A	Access Bits	Key B
FFFFFFFFFFFF	FF078069	FFFFFFFFFFFF

MFOC

2008
https://github.com/nfc-tools/mfoc

MFCUK

2010
https://github.com/nfc-tools/mfcuk

PWND

Objetive

Given:

Usually, only UID is used
Bloc 0 no writable, default key
Chinese clones allow writing to bloc 0

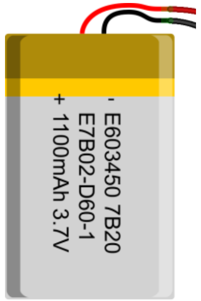
I’m going to:

Prove how **EASY** is to **stole and Identity**

Objetive

RFID Reader / Writer / Cloner

- Autonomous
- Portable
- Non volatile
- Remote control



Objetive

RFID Reader / Writer / Cloner

Automous + Portable + Non volatile

ESP8266 (NodeMCU Dev Kit)

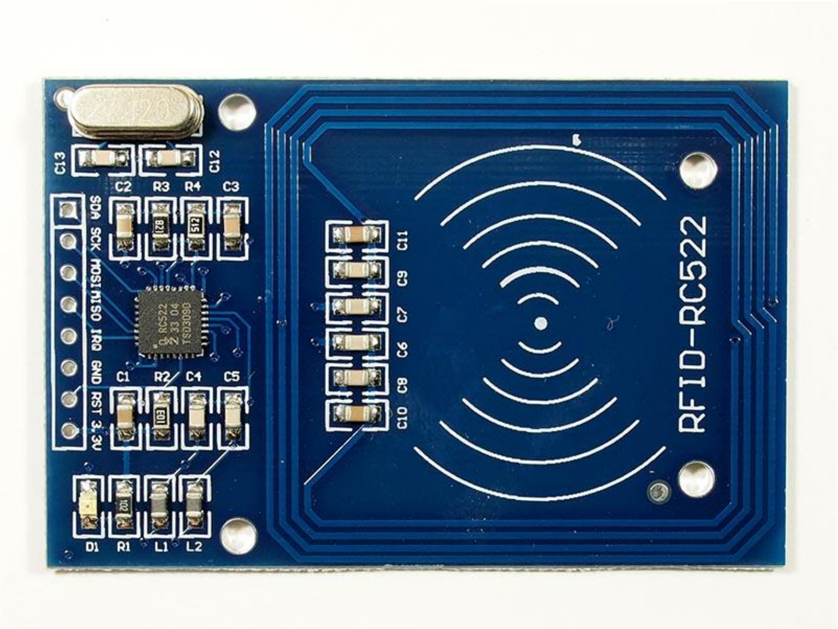
- Low cost (<3€)
- Small factor
- Wifi
- SPI
- Filesystem (Non volatile)
- Plenty of community libraries

Objetivo

RFID Reader / Writer / Cloner

Automous + Portable + Non volatile

MFRC522 (RFID Reader/Writer)



- Low cost (<2€)
- Small factor
- SPI communication

Objetivo

RFID Reader / Writer / Cloner

Remote control

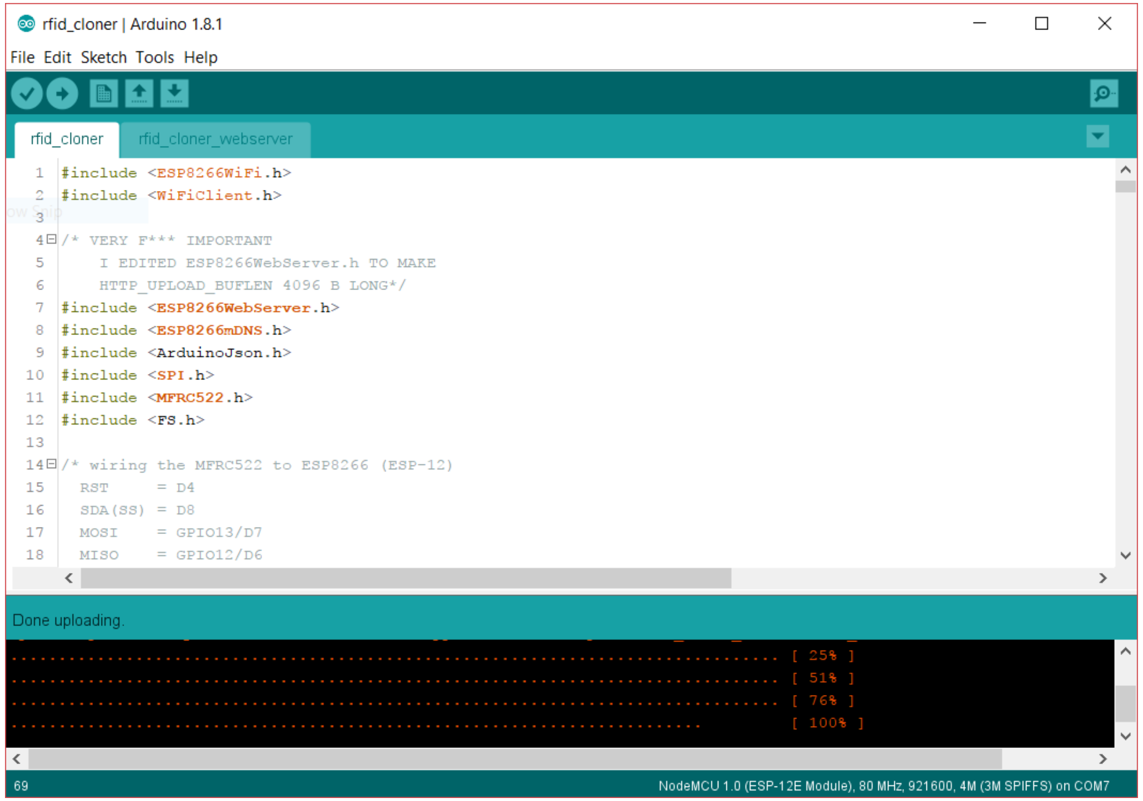
Android Smartphone

- Bajo coste
- Wifi/Bluetooth
- Interfaz tactil
- Facil de programar
- ¿Quien no tiene uno?

Implementation

ESP8266 + RC522

Arduino IDE



```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>

/* VERY F*** IMPORTANT
    I EDITED ESP8266WebServer.h TO MAKE
    HTTP_UPLOAD_BUFLEN 4096 B LONG*/
#include <ESP8266WebServer.h>
#include <ESP8266mDNS.h>
#include <ArduinoJson.h>
#include <SPI.h>
#include <MFRC522.h>
#include <FS.h>

/* wiring the MFRC522 to ESP8266 (ESP-12)
  RST     = D4
  SDA(SS) = D8
  MOSI    = GPIO13/D7
  MISO    = GPIO12/D6
```

Done uploading.

```
.............................................................................. [ 25% ]
.............................................................................. [ 51% ]
.............................................................................. [ 76% ]
......................................................................         [ 100% ]
```

69 NodeMCU 1.0 (ESP-12E Module), 80 MHz, 921600, 4M (3M SPIFFS) on COM7

Libraries

ESP8266

Comunity
https://github.com/esp8266/Arduino

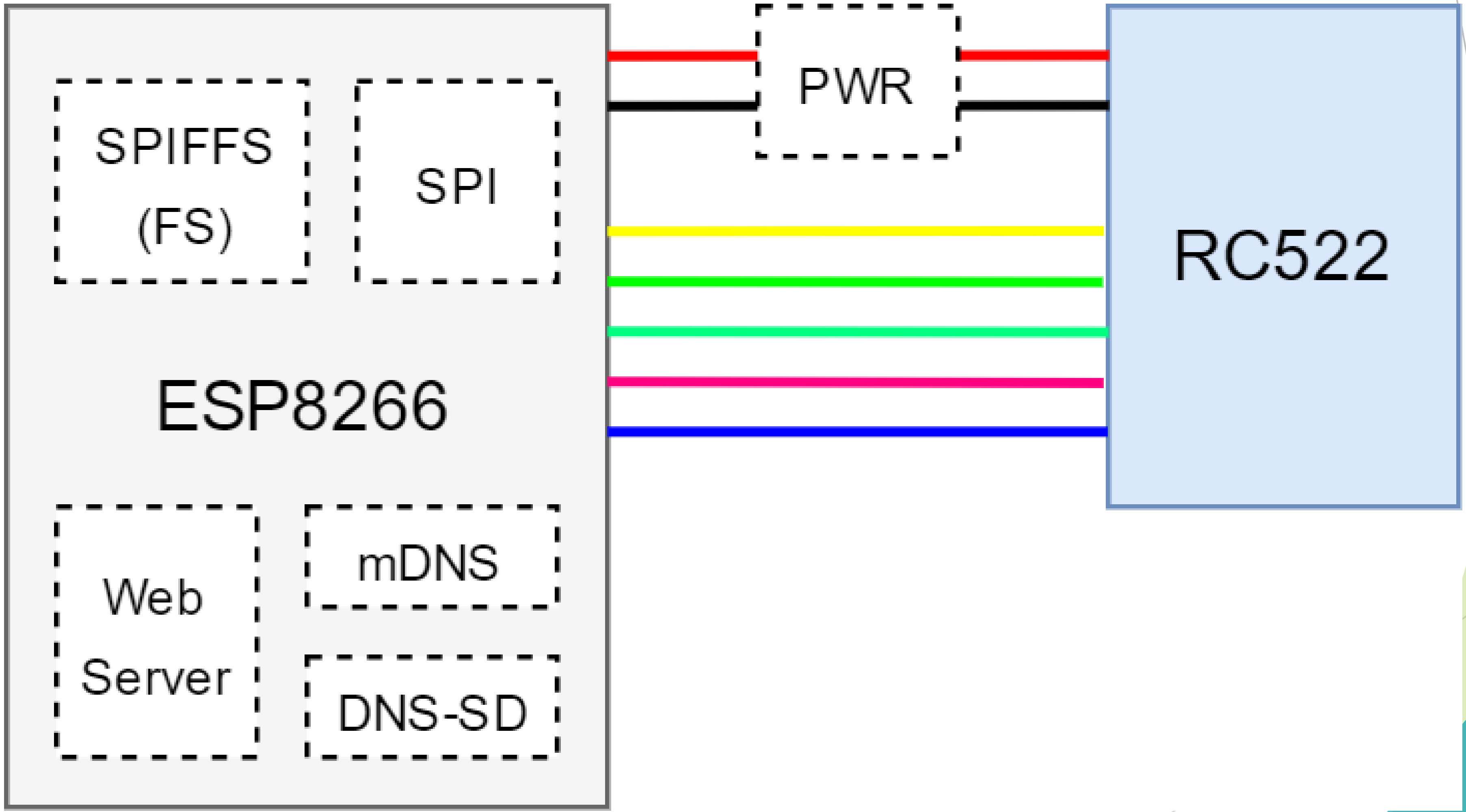
WebServer
SPIFFS
mDNS

RC522

Miguel Balboa
https://github.com/miguelbalboa/rfid

Implementation

ESP8266 + RC522



Implementation

ESP8266 + RC522

Promiscuous mode

Read and save all cards in rage

SPIFFS *(lib)*

3M

https://github.com/pellepl/spiffs

No directories, flat FS, directories = filters

Web Server

Wifi *(lib)*

mDNS responder *(lib)*

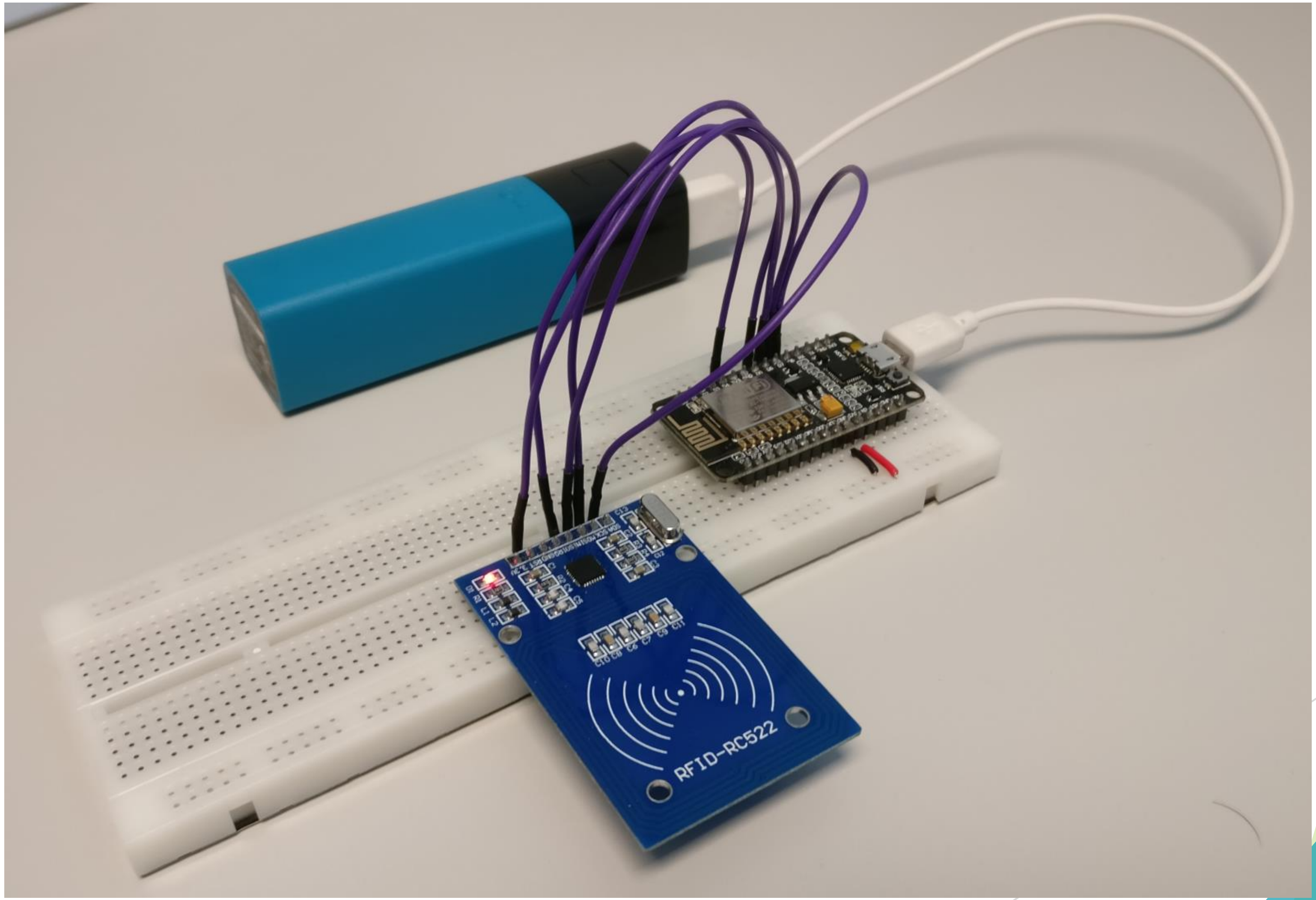
DNS-SD service advertiser *(lib)*

WebServer *(lib)* HTTP API

METHOD	URL	Descripción
GET	/cardslist	JSON cards list
GET	/card	Card in JSON format
DELETE	/card	Erase card
PUT	/card	Send cards, *write* param to write card

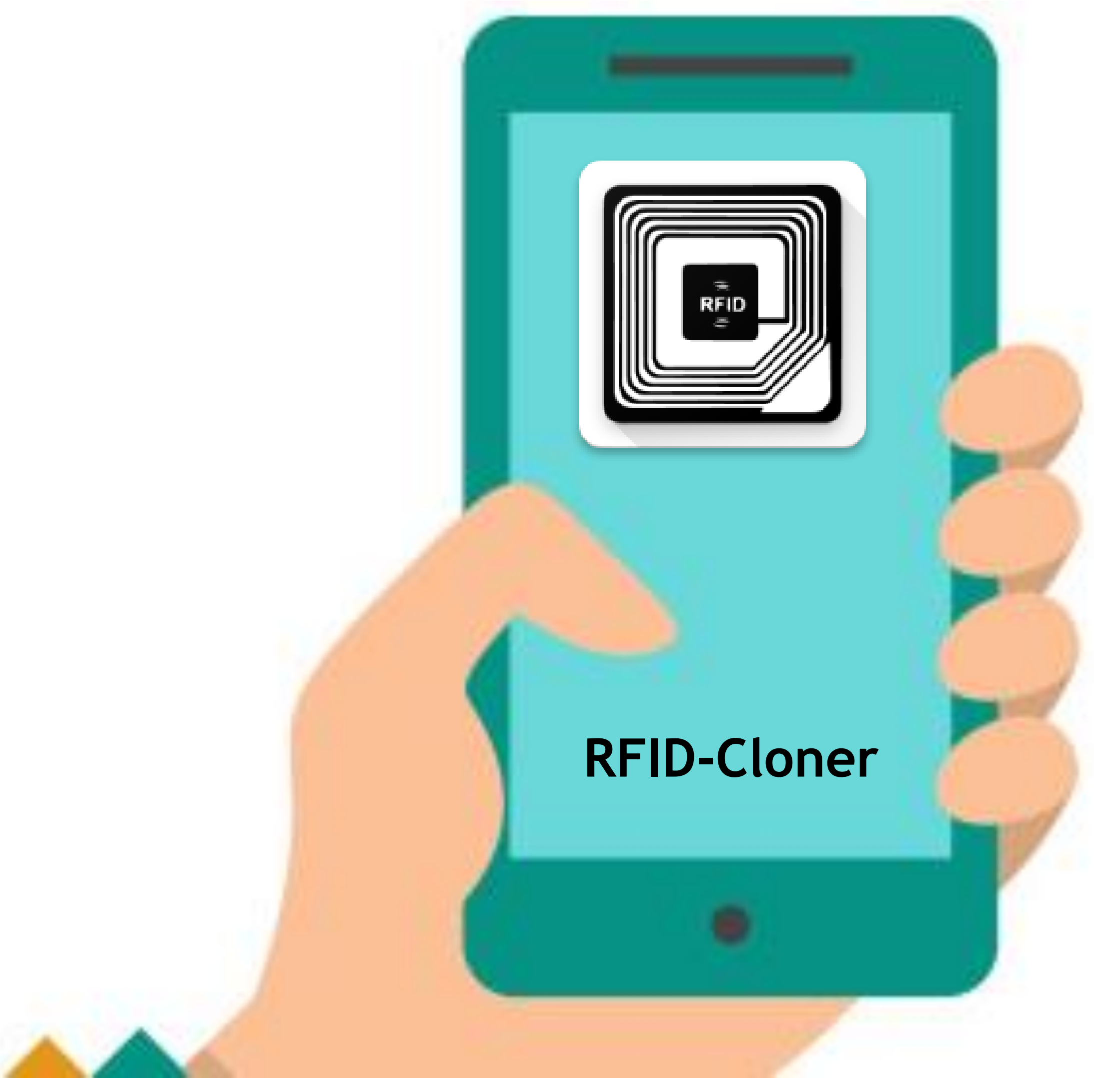
Implementation

ESP8266 + RC522



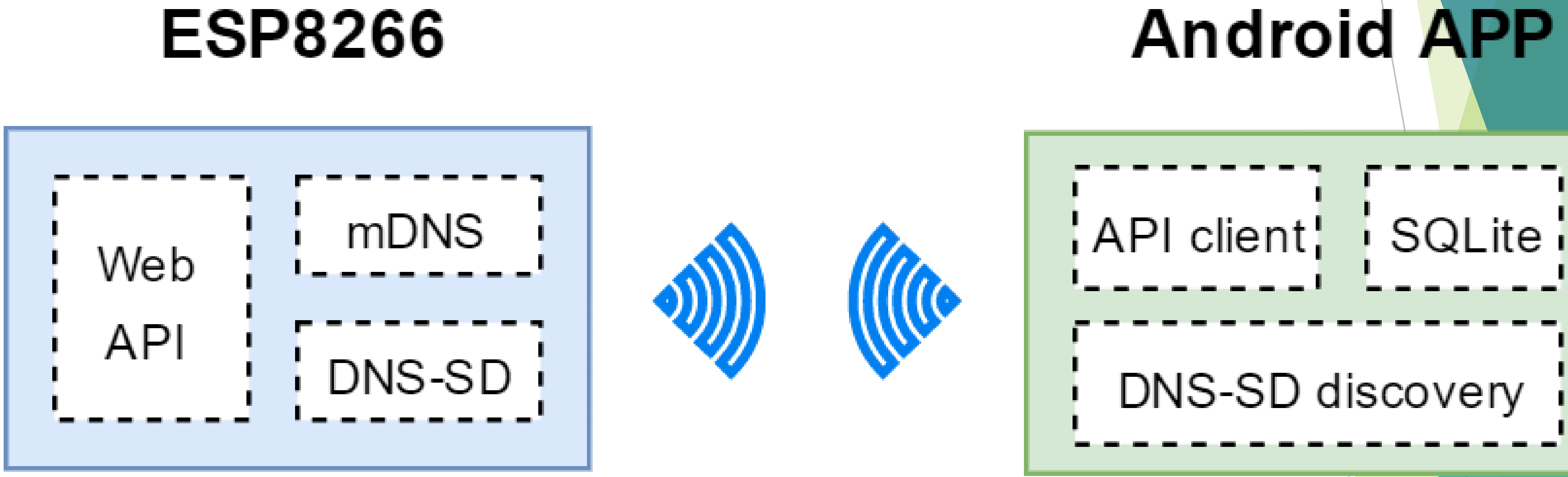
Implementation

Android APP



Implementation

Android APP


ESP8266

Web API

mDNS

DNS-SD

Android APP

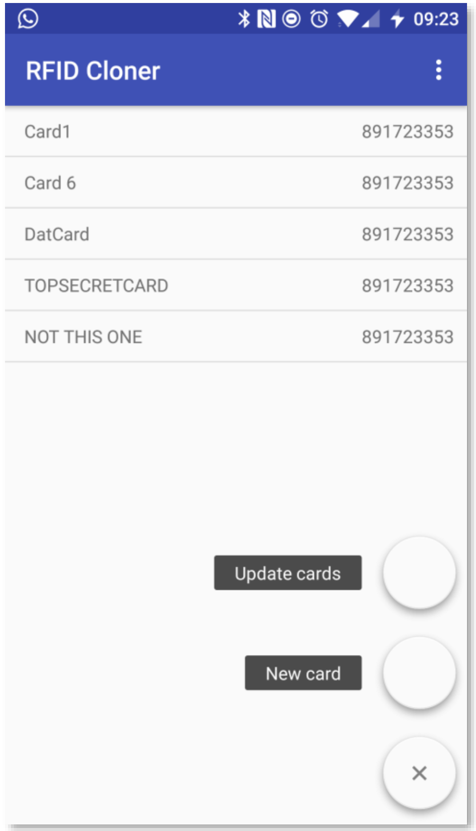
API client

SQLite

DNS-SD discovery

Implementation

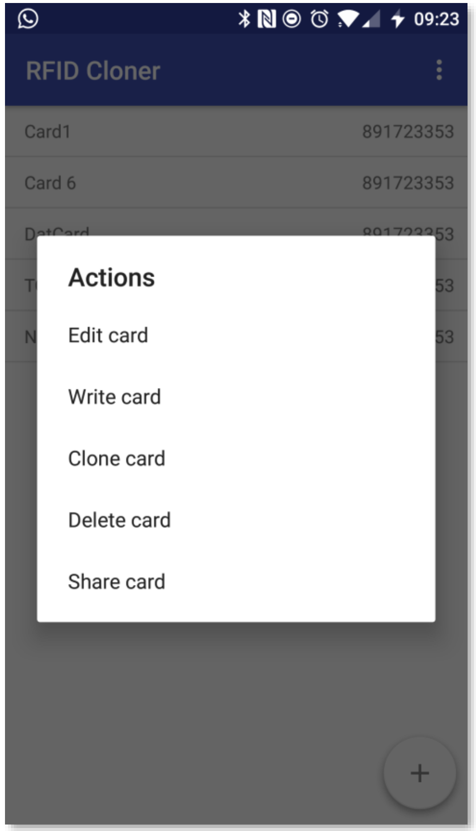
Android APP


09:23

RFID Cloner

Card1	891723353
Card 6	891723353
DatCard	891723353
TOPSECRETCARD	891723353
NOT THIS ONE	891723353

Cards list

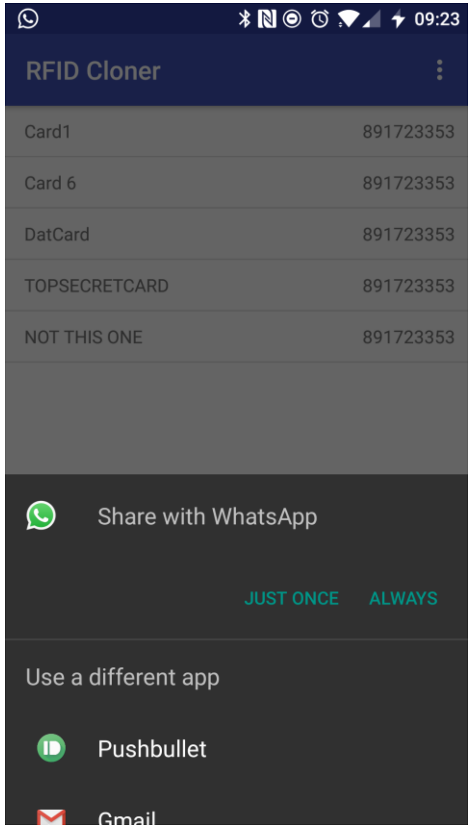

09:23

RFID Cloner

Actions

- Edit card
- Write card
- Clone card
- Delete card
- Share card

Actions over cards


09:23

RFID Cloner

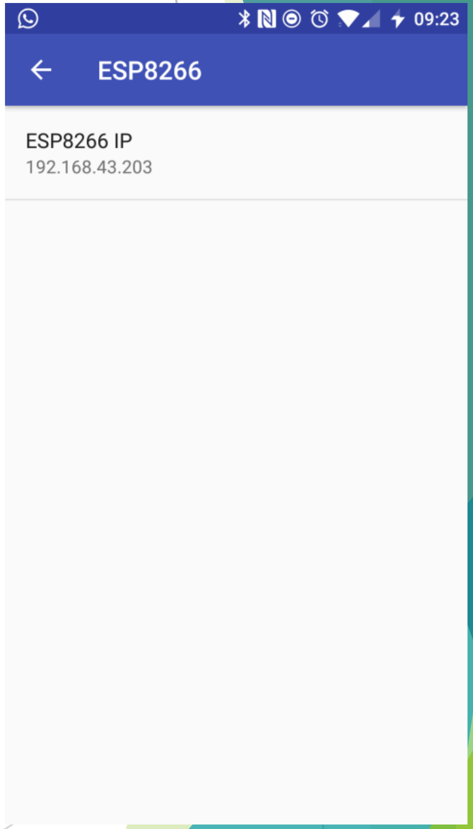
Card1	891723353
Card 6	891723353
DatCard	891723353
TOPSECRETCARD	891723353
NOT THIS ONE	891723353

Share with WhatsApp

Use a different app

- Pushbullet
- Gmail

Share cards


09:23

ESP8266

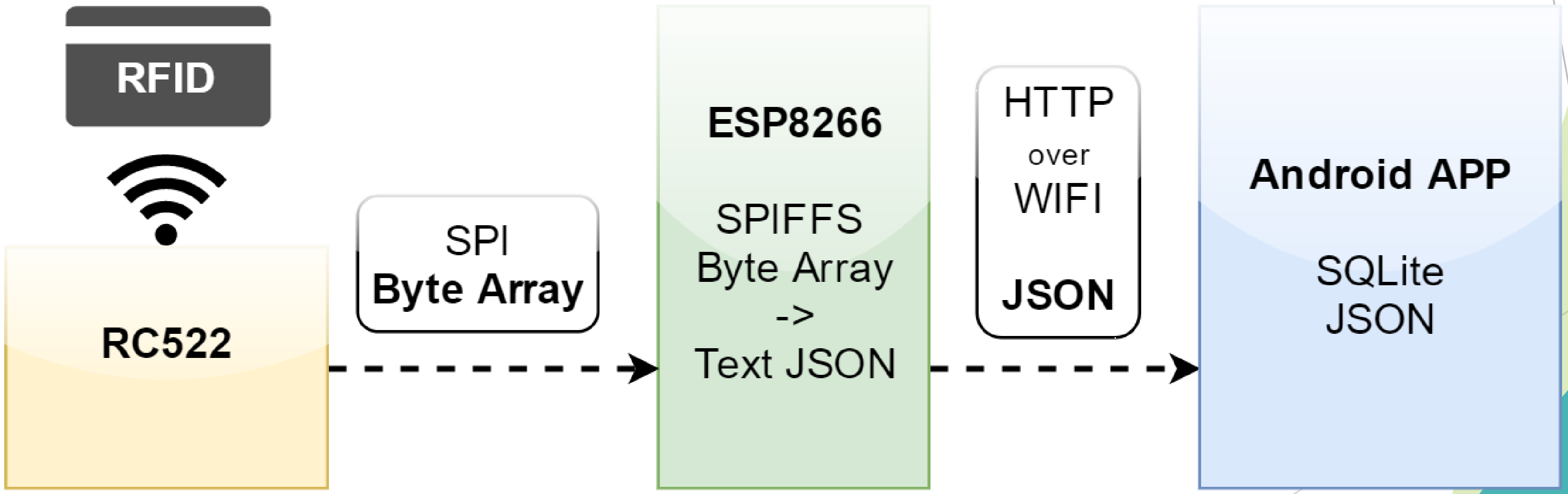
ESP8266 IP
192.168.43.203

mDNS/DNS-SD fallback

Implementation

ESP8266 + RC522 + Android APP

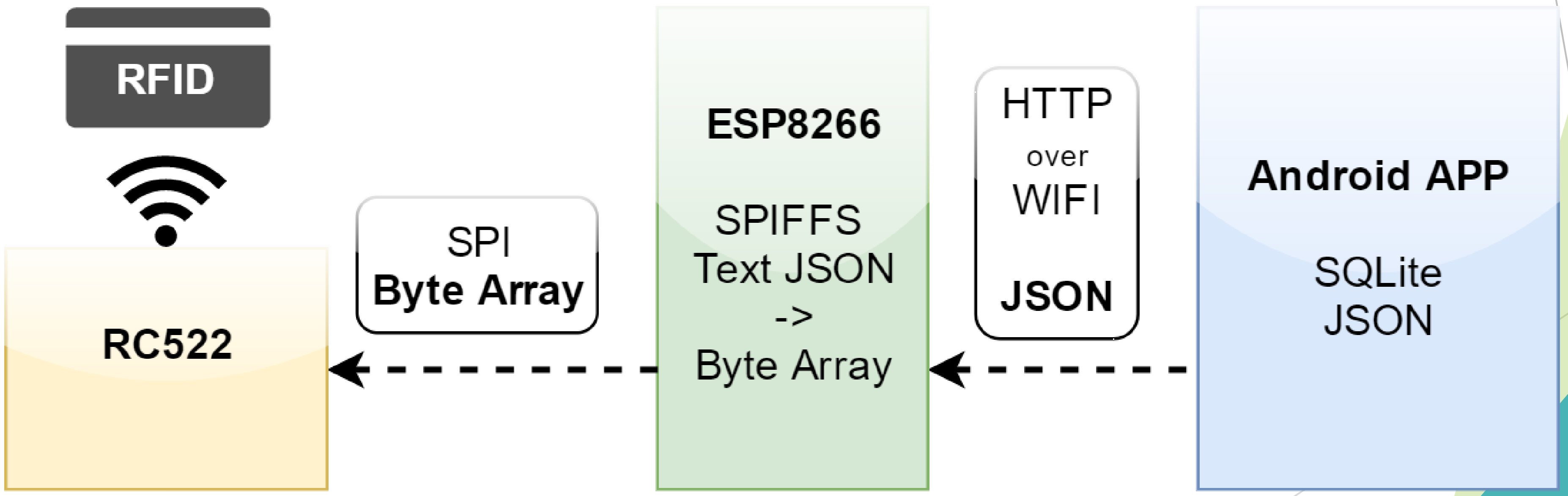
READING



Implementation

ESP8266 + RC522 + Android APP

WRITING


RFID

RC522

SPI
Byte Array

ESP8266

SPIFFS
Text JSON
->
Byte Array

HTTP
over
WIFI

JSON

Android APP

SQLite
JSON

SHOW
TIME!

Q&A

AND EVALUATION