NFC Terminal Prototype

Project Roadmap, High-level Design and Requirements

Point of contact → dotStar Technologies B.V.
Zuid-Hollandlaan 7, 2596 AL, Den Haag, Netherlands
Email: alexander.hoffmann@dotstar.tech

01

Objectives, Purpose & Scope

Main objectives of the given project is to design, develop and manufacture a prototype device with NFC read-write capabilities and a touch display. The device is intended to be used as an access control terminal by air traffic controllers (ATC) inside facilities like airport towers and control centers of the german air traffic management organization (DFS) https://dfs.de/homepage/en/. The primary use case assumes interaction between the terminal and the user's personal NFC tag, i.e. corporate ID card, to authenticate the user first on the backend Identity and Access Management (IAM) system and then to perform the user authorization, i.e. to check if the given user has enough permissions to execute ATC roles he selected on the touch screen. In the described scenario, both authentication and authorization are remote calls to the backend infrastructure.
Beside the primary goal to be an access control terminal, the prototyped device has a secondary usage purpose to perform background voice/sound recording in the air traffic control rooms. Despite the fact that this design goal is not considered to be the primary one, it might be the decisive one for acceptance success by the main customer (DFS).

In addition to the pure mechanical and hardware prototyping, design and development of the operating system, firmware and lifecycle management infrastructure are important parts of the project. Requirements and related design decisions for all parts of the project will be detailed in the 'Requirements and High-level Design' section.

The ultimate goal of the project would be to manufacture 3-5 NFC terminals provisioned with an engineering version of operating system and firmware and integrated with the lifecycle management backend. Produced prototypes will be aimed for a CE certification for office devices, whereas the certification campaign itself is out of scope of the prototyping phase.

Considering very ambitious technical goals and very tight time and budget constraints of the prototype, the requirements will be labeled as mandatory or optional providing some flexibility on the final deliverables.

The desired outcome of the requirements analysis phase is the technically optimal but yet viable and cost-effective solution with a trade-off analysis against other possible solutions.

02

Vision

The prototype under design is by far not limited by its primary purpose to be an access control terminal for air traffic controllers. A broad spectrum of similar use cases for instance for

- Time management
- Resource booking, e.g. hotel rooms
- Payment transactions, e.g. tram/train rides

can be easily facilitated by the prototyped NFC terminal. In some use cases the user's cell phone can perform the role of a personal NFC tag.

As long-term objective we envisage the following two scenarios:

- Providing the terminal possibly for free in the frame of service subscription
    - Managing full lifecycle of the terminal
        - Operating system and firmware updates
        - User’s business logic updates
        - Device repair and renewal
    - Charging customers based on transactions/traffic level
- Selling the terminal hardware with maintenance support to cover the production costs plus some margin

03

Requirements & High-level Design

Entire work to design and develop the NFC prototype can be splitted in the following four domains:

- 3D modeling and printing of the enclosure
- Design and manufacture of hardware and custom PCB
- Design and development of the operating system and firmware
- Design and development of a lifecycle management infrastructure

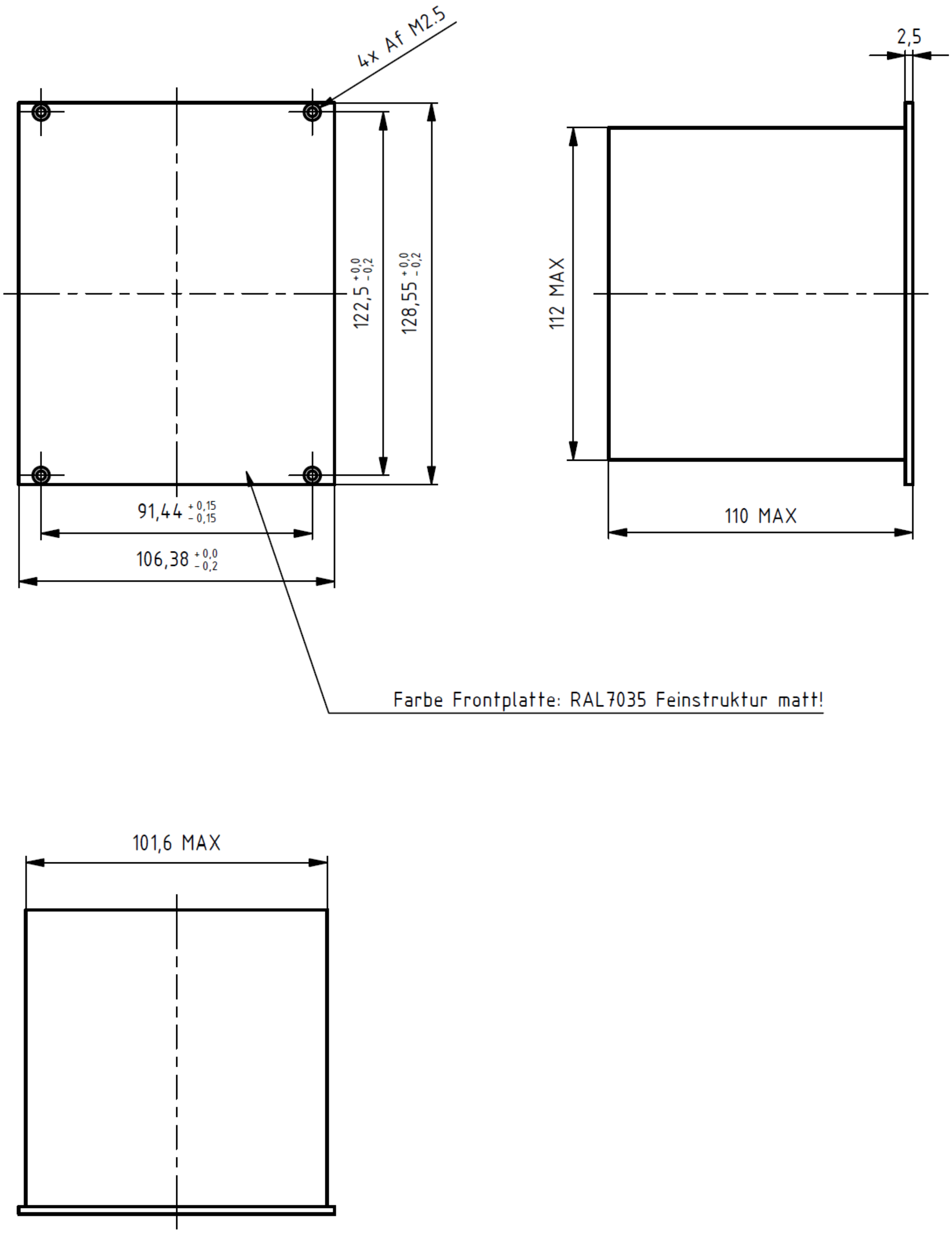
Requirements and high-level design concepts are represented in sections below.

04

Enclosure

3D models design of the terminal's enclosure shall be done in some commonly used CAD application and delivered at least in STP format. Enclosure dimensions can be taken from the picture below, however can deviate slightly from the given values in the range:

- Width 105mm - 107mm
- Height 128mm - 129mm
- Depth 110mm max (the less the better)
- Enclosure (back part):
    - Width 102mm max (sketch below depicts 101,6, which can be rounded up to102)
    - Height 112mm max


4x Af M2.5
2,5
122,5 +0,0 -0,2
128,55 +0,0 -0,2
112 MAX
91,44 +0,15 -0,15
106,38 +0,0 -0,2
110 MAX
Farbe Frontplatte: RAL7035 Feinstruktur matt!
101,6 MAX

All final prototypes shall be provided fully assembled with the enclosure. Whereas the enclosure shall be printed by the Selective Laser Sintering (SLS) or Multijet Fusion process using the PA12 material.

The two following 3D printing service has been proven to be quite reliable and cost-efficient:

- https://www.sculpteo.com
- https://canto.jellypipe.com

05

Hardware

Targeted hardware architecture is a subject to trade-off analysis and detailing in the frame of the given project, however the hard requirements, which the prototype cannot deviate from are listed below

- ARM CPU architecture
- NFC tag read-write capabilities (preferably NXP PN7160 board)
- GPIO header with I2C support
- LCD Touch screen
- Power over Ethernet (PoE)
- Real Time Clock (RTC)
- Integrated Microphone
- eMMC with preferably 8G (more possible but not desirable)
- Memory 128 MB to 2GB (less is better)
- Hard-reset button

System on Module (SoM) is the most important hardware part, which will define almost the entire design later in the project. Decision on the SoM to use for the prototype shall take into account the following trade-offs:

- Off-the-shelf I/O boards vs a custom I/O board design
- Firmware/software support
- Mass provisioning approach

One of the obvious candidates for the SoM would be a CM4 (version CM4001008 without WiFi, 8GB eMMC, 1GB RAM). Its pros against other alternatives are a great eco-system of software like operating system, provisioning server, huge community, good documentation and many others. However there are two main cons as well:

- Relatively high price (40 euro CM4 + at least 20 euro I/O board)
- Not a great choice of available off-the shelf I/O boards considering the required dimensions of the enclosure

A good, affordable and tested with previous prototypes I/O board for the CM4 SoM is a Nano Base Board (B) by Waveshare. However the Nano BAse Board (B) is missing such crucial components as RTC and PoE.

Another option for the hardware platform to use is the Luckfox Pico Ultra RV1106. It represents an integrated board with

- Single-core Cortex-A7 CPU (Rockchip RV1106G3)
- 8GB EMMC
- 256MB DDR3L
- PoE module
- RGB LCD Interface
- Onboard microphone
- GPIO header with I2C support
- RTC battery connector
- Reset button

Despite all the advantage of the Luckfox Pico Ultra board there some crucial down sides as well:

- Relatively weak community, documentation and software support
- Outdated 32-bit CPU architecture (ARM Cortex-A7)

Luckfox provides another Linux development board [Lyra Ultra RK3506B](). It might be considered instead of the above Pico Ultra RV1106 as the hardware platform. However it is missing an integrated microphone and uses a DSI display interface (MIPI DSI 2-lane). Different LCD interface isn't necessarily a problem, but a compatible display has to be found e.g. https://www.waveshare.com/4inch-DSI-LCD.htm by Waveshare.

Considering LCD choices one has to take into account the following hard requirements:

- Enclosure dimensions (see above)
- Touch functionality
- High resolution (the higher the better)

A perfect option here would be 4 inch (720×720 Pixels) IPS capacitive touch display with RGB communication interface by Luckfox.

NFC board is another crucial hardware component which has to be integrated into the prototype. Among many possible options preference should be given to the boards with better firmware support provided as open source Linux user space library or a kernel module. A suitable NFC board is for instance NFC 7 Click - I2C by MikroE. Previous prototypes successfully integrated and tested a NFC 2 Click board which is using a discontinued NXP chip PN7150. It has been proven to be a reliable and easy to integrate solution. The newer NFC 7 Click board is using the PN7160 chip by NXP, which offers very similar firmware/driver support (master branch is for PN7150 and this branch contains support for the PN7160).

The biggest advantages of the PN7160-based boards are:

- Open source firmware/driver
- Great documentation
    - https://www.nxp.com/docs/en/application-note/AN12989.pdf

    - https://www.nxp.com/docs/en/application-note/AN13189.pdf
    - https://www.nxp.com/docs/en/user-manual/UM11495.pdf

Among the disadvantages of the NFC 7 Click board one can point to the

- Inflexible decision to integrate NFC antenna into the board
- Relatively high price

An alternative design might consider using a board similar to the NFC Extend Click with an attachable antenna. However firmware/driver support of such a solution has to be carefully evaluated.

Obviously the ultimate integration test of the prototype hardware would be to provision it with custom-built embedded Linux containing a test client app and run manually NFC transactions against the backend infrastructure. However acceptable integration test setup is allowed to run on a pre-built operating system for instance RaspberryOS, since building custom embedded Linux might be too difficult to complete in the frame of the given project.

06

Operating System & Firmware

A custom embedded Linux based on Yocto or Buildroot project shall be designed for the prototype. Since this activity strongly depends on the hardware architecture, it is acceptable to use a pre-built Linux image during the development phase. However, regardless of whether a custom or a pre-built image is used, it has to provide kernel/firmware support for

- Touch LCD
- NFC sensor
- RTC

- PoE
- Microphone

Moreover it shall be bootable from the eMMC and follow the usual best-practices for embedded Linux development, e.g. minimal footprint, no package manager, no build toolchains, etc.

07

Infrastructure

Software infrastructure for the given prototype has to provide at least the operating system provisioning server. Such a server has to be able to perform mass provisioning of a Linux image to the terminals connected either by LAN or a USB cable. Similar solution is provided for RaspberryPi boards here with good documentation.

Beyond the mass provisioning server the infrastructure might provide the following services in order of importance (the higher the more important):

- MQTT v5 messaging broker with persistency support
  - Mosquitto MQTT broker with a custom libSQL plugin
  - TBMQ and ThingsBoard
  - VerneMQ and a custom plugin
- Automated Certificate Management Environment (ACME) to manage X.509 certificates for mTLS connections between terminals and the backend, e.g. HCP Vault
- Identity and Access Management, e.g. Keycloak
- Monitoring stack, preferably based on Grafana/Prometheus/Loki

08

Integration and Performance Testing

A client user applications with a basic but neat and good looking UI to support the following two use cases:

- DFS-tailored Air Traffic Controller (ATC) roles bookings
- Simple check-in/check-out (preferably in C or Golang)

Beyond the actual client to be installed on the prototype hardware, an emulator (possibly headless without UI) might be very useful for performance testing with the following scenario - a number of Docker or LXC containers with the business logic of a real client shall be started in headless mode and connected to the messaging backend. Performance and resilience for instance of a 10K fleet of devices can be tested then without the real hardware.

09

Budget, Agreements & Timeline

A budget of 24K euro is allocated for the first 3 months of the design and development phase starting from Aug 2025. Additional contingency budget of up to 8K euros is allocated for Nov 2025.
Contractual work is going to be performed on time & material basis and paid with hourly rate 100 euro/hour (excluding VAT if applicable). Contractor will be requested to submit invoices each month stating the number of hours spent. Payment transaction will be made within 14 days after the invoice reception.
All agreed deliverables shall be made available in project repositories specified during the project.
Meetings to monitor progress and synchronize development teams shall be held on a weekly basis.