UNIVERSITY OF CAGLIARI

DIEE - Department of Electrical and Electronic Engineering

Lab8. Smart Fridge

Michele Nitti
michele.nitti@unica.it

Smart Fridge – Reference Scenario

Scenario Definition

- We are a company aiming to create a device to make refrigerators smarter, starting from an old refrigerator.

Objective: Get Rich!

- Assumptions
  - We are in a home environment with WiFi connectivity. We are dealing with a non-IoT refrigerator to which we want to add a device to enhance its capabilities. The functional requirements in the following slide explain these capabilities.
  - The refrigerator is used by the entire family.

Smart Fridge

- Reminder! All the projects must follow the following requirements:
  - All systems must be IoT
    - Data has to be available for anyone or anything requesting it
  - Data produced by physical devices must be accessible at any time
  - Physical devices must be considered battery-powered (even if during the prototype they are not!)
  - All projects must be designed taking into account the possible evolutions of the project itself

Smart Fridge – Requirements

Functional Requirements
FR1 - Fridge Open Monitoring **Requirement**: The system is able to monitor whenever the fridge is opened. **Input**: Fridge door status **Output**: System status update
FR2 - Open Duration Notification **Requirement**: The user can receive a notification, both as a message and a sound, when the fridge has been open for more than K seconds. **Input**: Trigger regarding the opening status of the refrigerator **Output**: Notification (message and sound)
FR3 - Temperature Monitoring **Requirement**: The system is able to monitor the temperature of the fridge. **Input**: Current fridge temperature **Output**: System status update
FR4 - Temperature Chart Request **Requirement**: The user is able to request a chart that represents the temperature of the fridge over time. **Input**: User request for temperature chart **Output**: Chart representing the temperature of the fridge over time

Smart Fridge – Requirements

Functional Requirements
FR5 - Total Open Time Request **Requirement**: The user can require the total opened time during the last hour/day/week. **Input**: User request for total open time (last hour/day/week) **Output**: Total time the fridge was open in the requested period
FR6 - User Login and Registration **Requirement**: Users can sign in or sign up for the system. **Input**: User credentials **Output**: User login status/confirmation of new user creation

Non-Functional Requirements
NFR1 - Affordable: The device must be cost-effective to attract a wide range of customers.
NFR2 - Low Battery Consumption: The device must have minimal power consumption to ensure long battery life.
NFR3 - Real-Time Notifications: The device should be capable of sending real-time notifications to inform users about any updates or issues.

Smart Fridge – What do we need?

- **A device to monitor the fridge**
  - We aim to develop a device that can monitor various aspects of a refrigerator's operation, such as door status and internal temperature. This device will be designed to work with non-IoT refrigerators, effectively turning them into smart appliances.
- **Choosing a Controller**
  - The choice of controller is crucial as it will drive the capabilities and performance of our device. Factors to consider include processing power, energy efficiency, cost, and compatibility with other hardware components.
- **Sensors and Actuators**
  - Door Status Sensor: This sensor will detect when the fridge door is opened or closed.
  - Alert Sound Actuator: This component will produce an audible alert when certain conditions are met, such as the fridge door being left open for too long.
  - Temperature Sensor: This sensor will measure the fridge's internal temperature. It must be accurate and reliable, as incorrect temperature readings could lead to food deterioration.

Smart Fridge – What do we need?

- **User Interface**
  - Who is the customer? Our primary customers are homeowners who want to enhance the functionality of their existing refrigerators. This includes families, individuals living alone, and anyone else who uses a refrigerator in their home.
  - Which interface is the most user-friendly? Given our target customers, a user-friendly interface would be one that is intuitive and easy to navigate.
- **Application Protocol**
  - Application protocols significantly impact both functional and non-functional requirements. They define how the device communicates with other devices or systems, which can affect performance, power consumption, and other factors.
  - Given that our device is battery-powered, we need to choose a protocol that is energy-efficient. This means it should allow the device to enter a low-power mode when not in use and minimize the amount of data transmitted to save power.

Smart Fridge – Controllers

Arduino

Feature	Rank
Cost	Ok but.. The WiFi shield rises the cost (~~NFR1~~)
Consumption	Ok
Computational Power	Ok

NodeMCU

Feature	Rank
Cost	Ok
Consumption	Ok
Computational Power	Ok

Raspberry

Feature	Rank
Cost	No (~~NFR1~~)
Consumption	No (~~NFR2~~)
Computational Power	Ok but not required

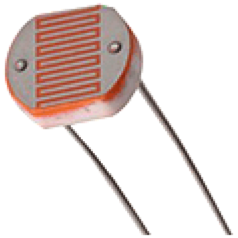
Others?

Smart Fridge – Requirements

FR1: “The user is able to monitor whenever the fridge is opened.”

- What sensors can be used to detect when the refrigerator door is opened?

- ❑Contact Sensors
- ❑Proximity Sensors: PIR or Ultrasonic sensor
- ❑Light Sensors: Photoresistor
- ❑Hall Effect Sensor
- ❑Others?



Smart Fridge – Requirements

FR3: “The system is able to monitor the temperature of the fridge.”

- What sensors can be used to measure the temperature?

Feature	DHT11	DHT22
Operating Voltage	3 to 5V	3 to 5V
Max Operating Current	2.5mA max	2.5mA max
Humidity Range	20-80% / 5%	0-100% / 2-5%
Temperature Range	0-50°C / ± 2°C	-40 to 80°C / ± 0.5°C
Sampling Rate	1 Hz (reading every second)	0.5 Hz (reading every 2 seconds)
Body Size	15.5mm x 12mm x 5.5mm	15.1mm x 25mm x 7.7mm
Advantage	Ultra low cost	More Accurate

Smart Fridge – Requirements

FR2: “The user is able to receive a notification, both as a message and a sound, when the fridge has been open for more than K seconds.”

FR4: “The user is able to request a chart that represents the temperature of the fridge over time.”

FR5: “The user can require the total opened time during the last hour/day/week.”

What User Interface?

- **Website**: A website can be accessed from any device with an internet connection, offering flexibility. However, it may not provide the same level of real-time interaction as a mobile application or Telegram bot.
- **Mobile Application**: A mobile application can provide a more personalised and interactive experience. It can send push notifications and allow users to access the temperature chart directly from their phones. However, it requires users to download and install the app and it is costly to implement.
- **Telegram Bot**: A Telegram bot can provide real-time interaction and easy accessibility. Users can request the temperature chart directly through their Telegram app. Moreover, creating a Telegram bot is free of charge, which can significantly reduce development costs. However, it requires users to have a Telegram account and may not offer as many features as a dedicated website or mobile application.

Smart Fridge – Requirements

- Application Protocols

ANY IDEA?

Smart Fridge – Requirements

Application Protocols

NFR3 – Real Time

ID	Functional Requirements	Protocol
FR1	The system is able to monitor whenever the fridge is opened.	HTTP
FR2	The user is able to receive a notification, both as a message and a sound, when the fridge has been open for more than K seconds.	MQTT + Webhooks/ HTTP
FR3	The system is able to monitor the temperature of the fridge.	HTTP
FR4	The user is able to request a chart that represents the temperature of the fridge over time.	Webhooks/ HTTP
FR5	The user is able to request the total opened time of the last hour/day/week.	Webhooks/ HTTP
FR6	User login and new user creation functionality	Webhooks/ HTTP

Smart Fridge – Sending Data

Which data does the fridge need to send?

How often?

- **Simple Approach**
  - In a simple approach, a system might send temperature data from the refrigerator at regular intervals.
  - This constant data transmission can consume a significant amount of battery power.
- **Proposed Approach: Data Burst Transmission**
  - Instead of sending data at regular intervals, we can send data in bursts.
  - This means the system would collect data over a period of time and send it all at once.
- **Advantages of Data Burst Transmission**
  - **Energy Efficiency**: Sending data in bursts can save battery power, as the system is not constantly using power to transmit data.
  - **Network Efficiency**: By reducing the frequency of data transmission, we can reduce network congestion and improve overall system performance.
  - **Reduced Overhead**: Each data packet sent over a network includes header information. By sending data in larger packets, less frequently, we can reduce the amount of header information sent and thus save additional power.
- **Considerations**
  - For our requirements, real-time temperature data is not necessary. Only alerts need to be real-time. This allows us to optimize data transmission without impacting the functionality of the system.

Smart Fridge – Sending Data First Scenario

- **Computation on the Device**

Pros:

- **Real-time processing**: Computation on the device can provide immediate feedback without network communication delay.

Cons:

- **Increased device resource usage**: Performing computations on the device can consume more processing power and battery life. This might not be ideal for devices with limited resources or applications needing energy conservation.

Smart Fridge – Sending Data Second Scenario

- **Computation on the Server**

Pros:

- **Conserves device battery**: Offloading computations to the server can help save battery life on the device.

Cons:

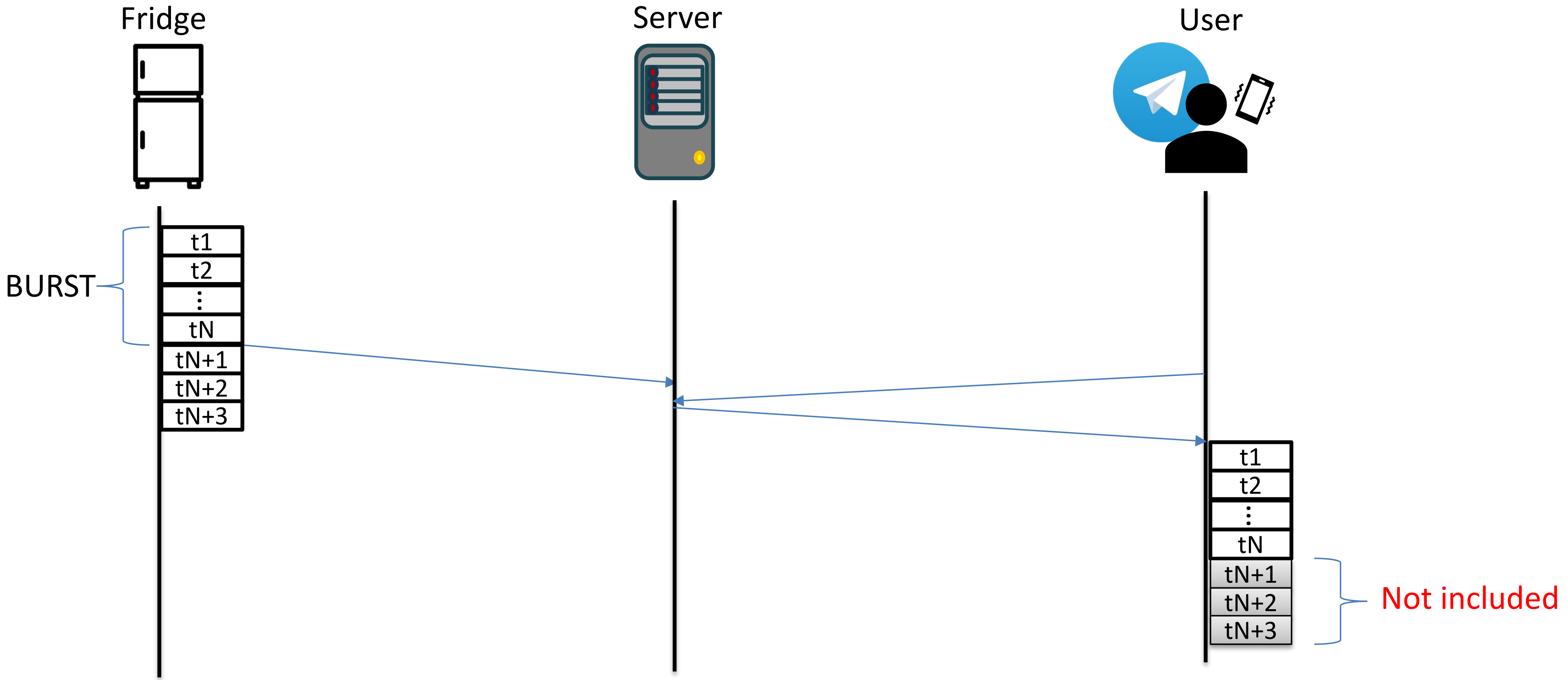
- **Notification delay**: There could be a delay in notifications. For instance, when the server calculates the time the fridge has been open, the user might have already closed the door. However, the notification could still arrive a few seconds later.

Smart Fridge – Sending Data Third Scenario with "Burst"

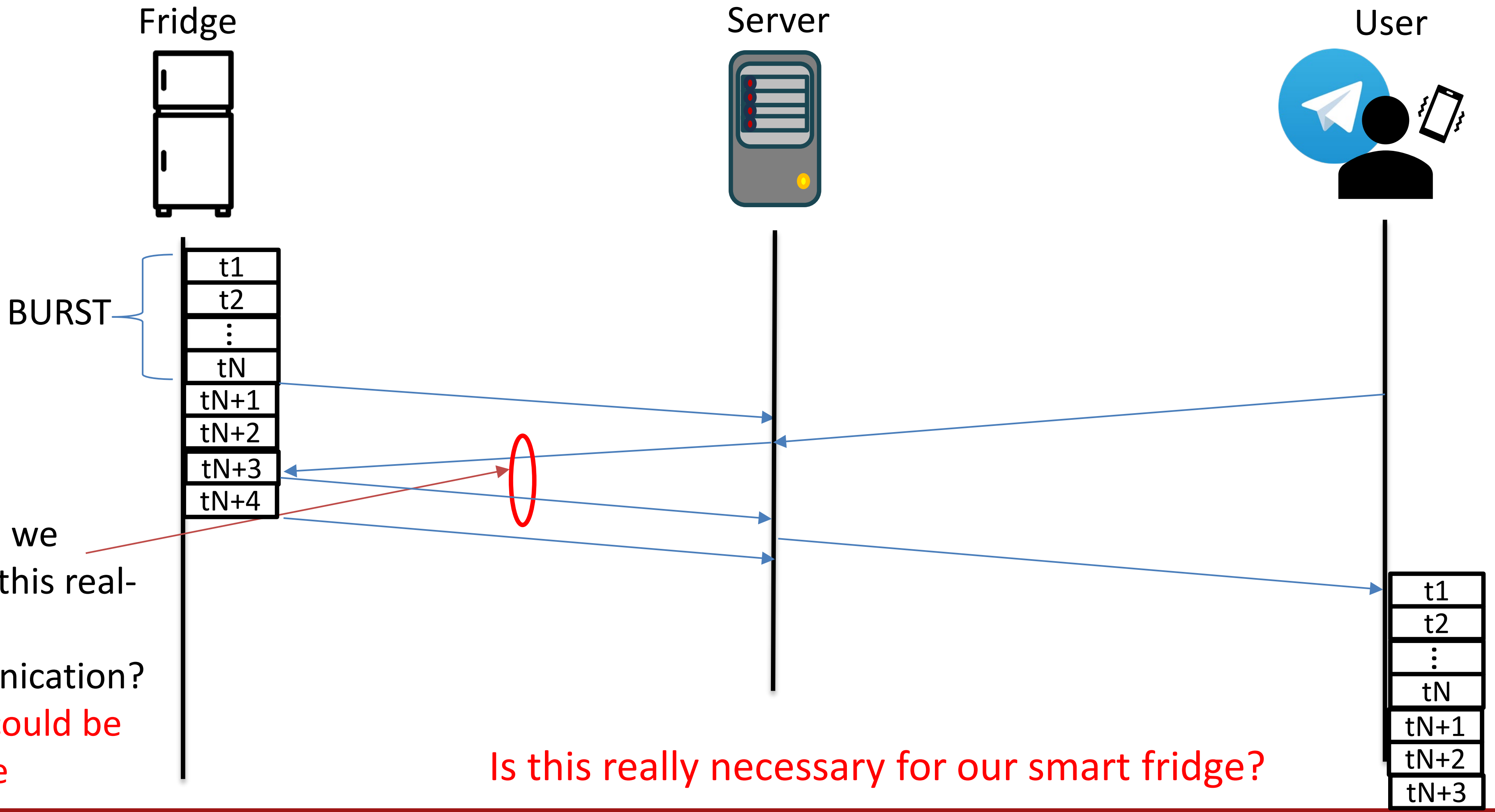
On-Device Computation and Temperature Chart Updates FR4

1. **Data Freshness**: If computation is performed on the device and data is sent in bursts every T* seconds, the temperature chart may not be up-to-date when the user requests. The chart will only reflect the data sent in the last burst.
2. **Missing Data**: Any measurements taken after the last burst and before the chart request will not be included in the chart. This could lead to gaps in the data presented to the user.
3. **Loopback Mechanism**: Implementing a loopback mechanism can mitigate this issue. This mechanism forces the device to send the collected data immediately upon a chart request, regardless of whether the elapsed time since the last burst has reached the time threshold.
4. **Up-to-date Charts:** The loopback mechanism will provide a way to include the most recent measurements, sending the user with the most up-to-date view of the temperature data.

Smart Fridge – Sending Data
Third Scenario with “Burst”



Smart Fridge – Third Scenario with "Burst" with loopback

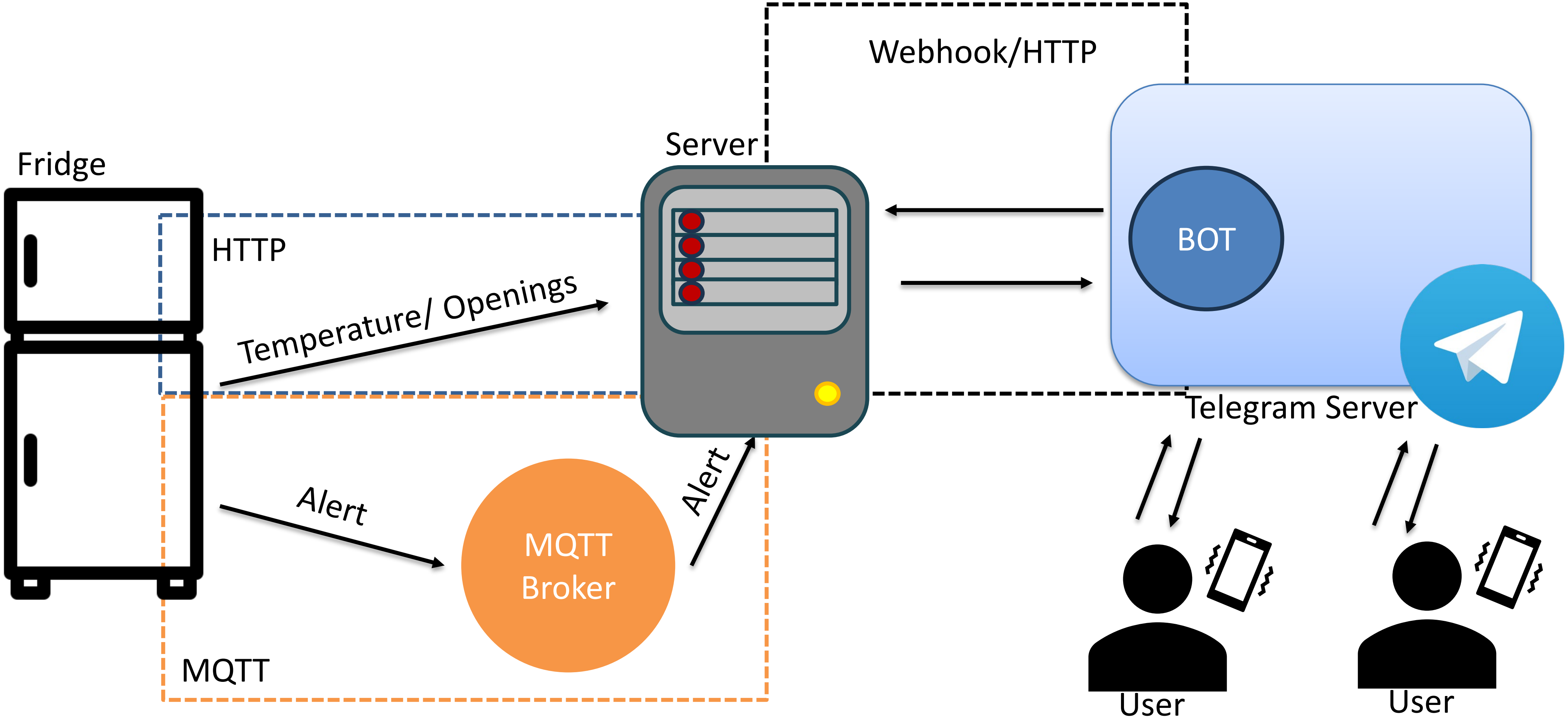


How do we handle this real-time communication?

MQTT could be a choice

Is this really necessary for our smart fridge?

Smart Fridge - Architecture



Smart Fridge – Digital Twins

Think about it…

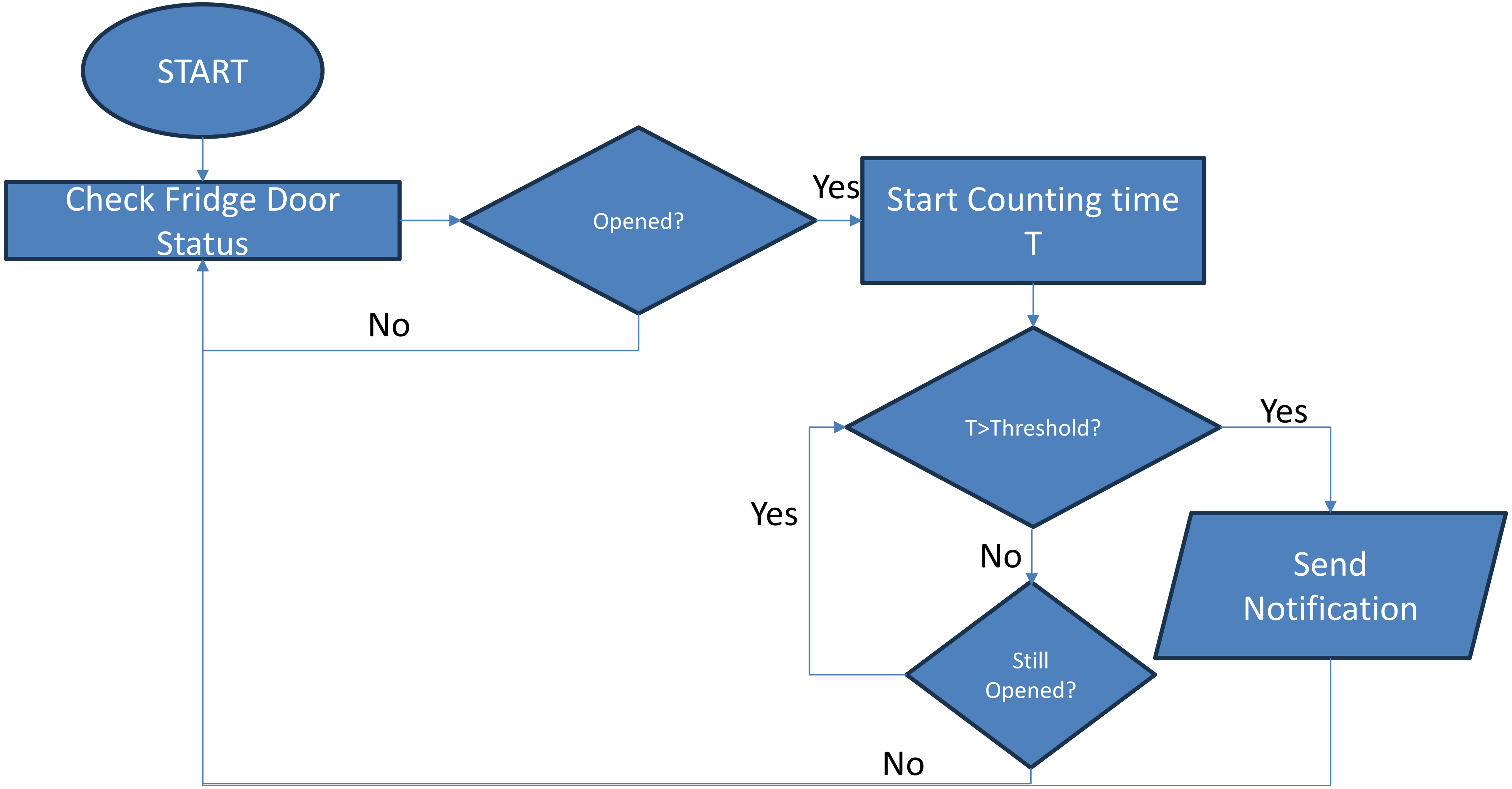
Smart Fridge – Digital Twins

- In our scenario, each refrigerator is represented as a digital replica, allowing us to monitor and control it remotely.
- Since the digital replica operates in the user space, both users and owners can interact with the refrigerator. User data is securely stored within the user space, enabling sign-in and sign-up functionalities.

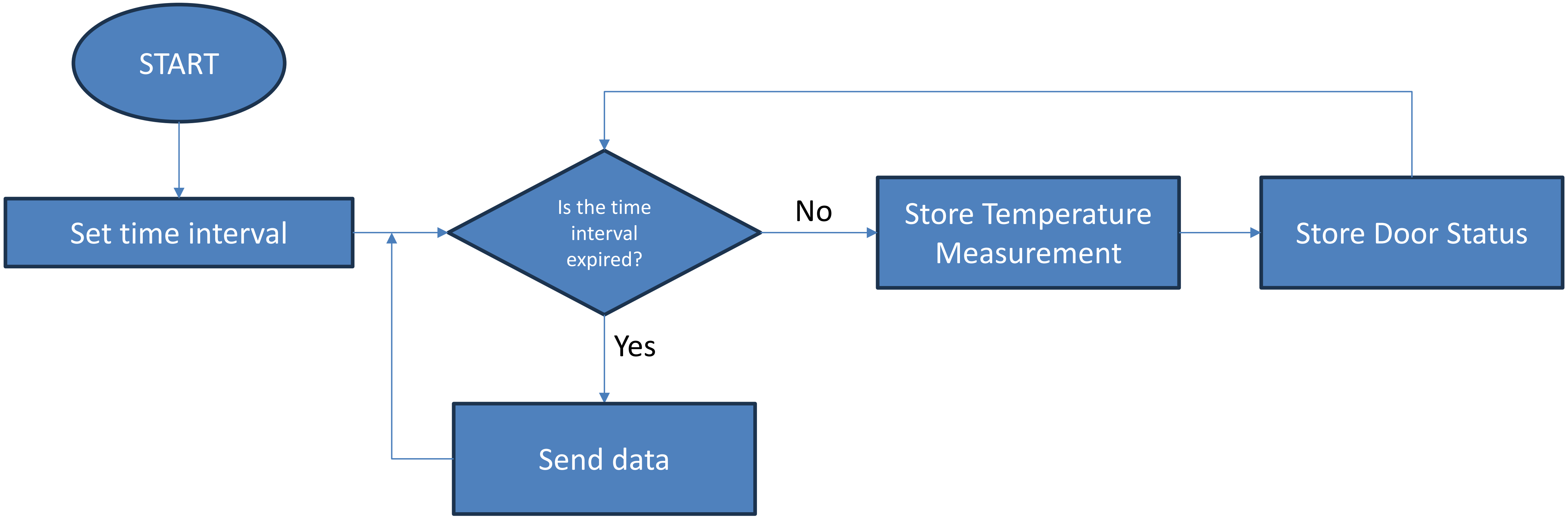
Smart Fridge – Digital Twins

- Who/ What is the Digital twin?
  - In such a simple scenario, it is difficult to identify who or what the DT represents.
- Evolutionary Scenario and future-proof design
  - Given that "all projects must be designed taking into account possible evolutions of the project itself", we could assume that the refrigerator might interact with the house in the future. This would create a **house DT**, where the refrigerator acts as its digital replica.

Smart Fridge – Door Checking (FR2)



Smart Fridge – Sending Data (FR1 and FR3)



Smart Fridge – Telegram Mock-Up

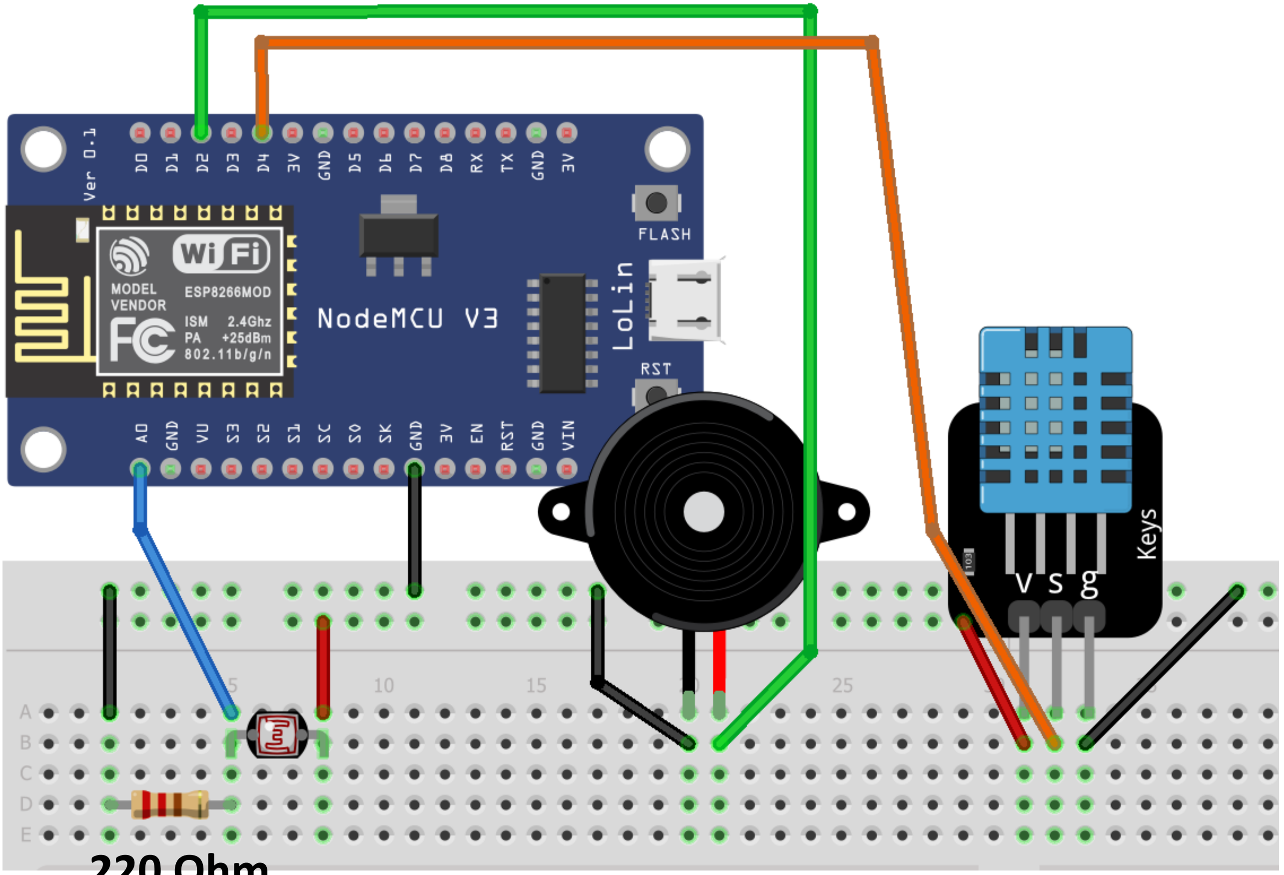
Unknown User

- /Register <Device_ID> <password> <name> FR6
- /Help → prints the menu command
- /Login <Device_ID> <password> FR6

Logged User

- /Logout → Necessary if the user wants to change refrigerator FR6
- /Help → prints the menu command
- /Chart → The User receives the temperature chart FR4
- /TotalTime → User request for total open time FR5

Smart Fridge – Wiring



NodeMCU	Photo Resistor
3.3 V	(+)
A0	(-)
NodeMCU	**Buzzer**
GND	(-)
D2	(+)
NodeMCU	**DHT**
3.3 V	V
D4	S
GND	G