

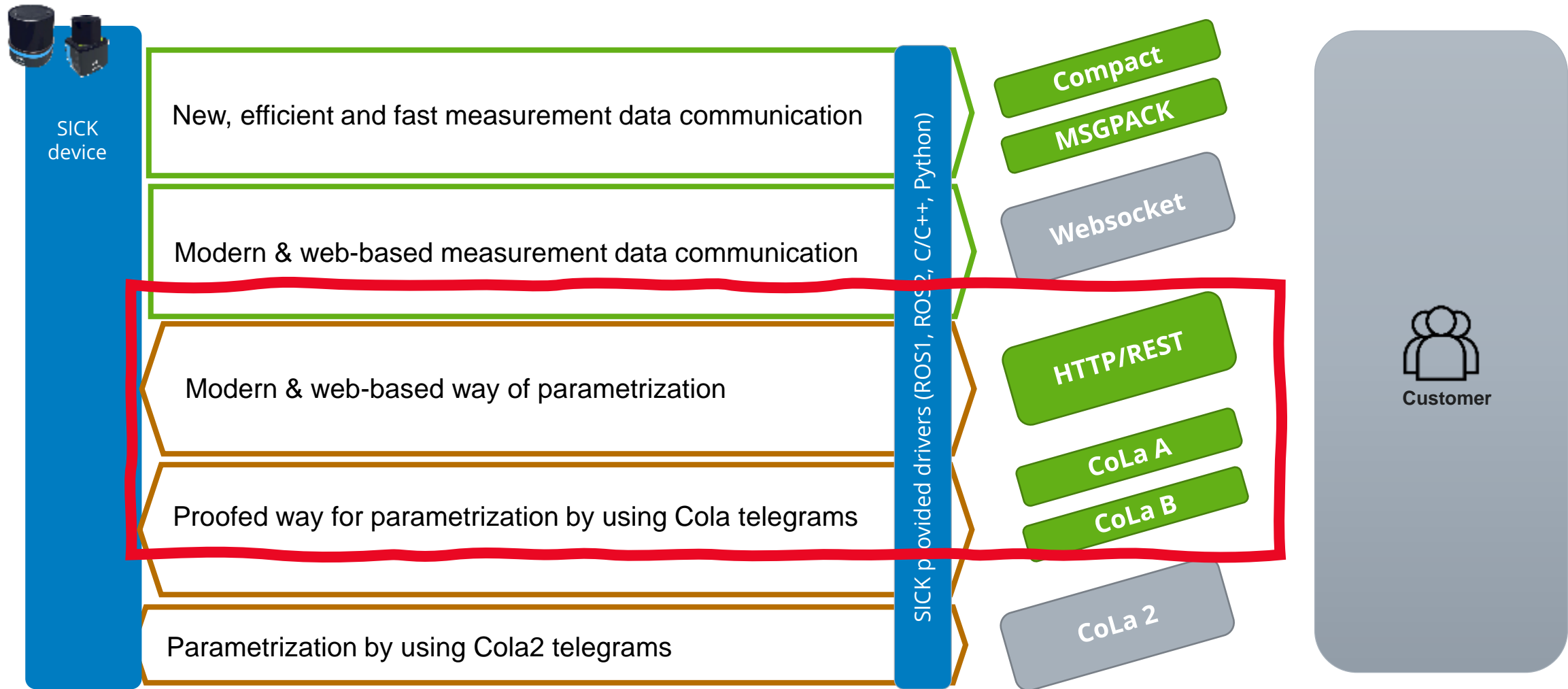
# Protocol and Integration: REST and OpenAPI



# Agenda

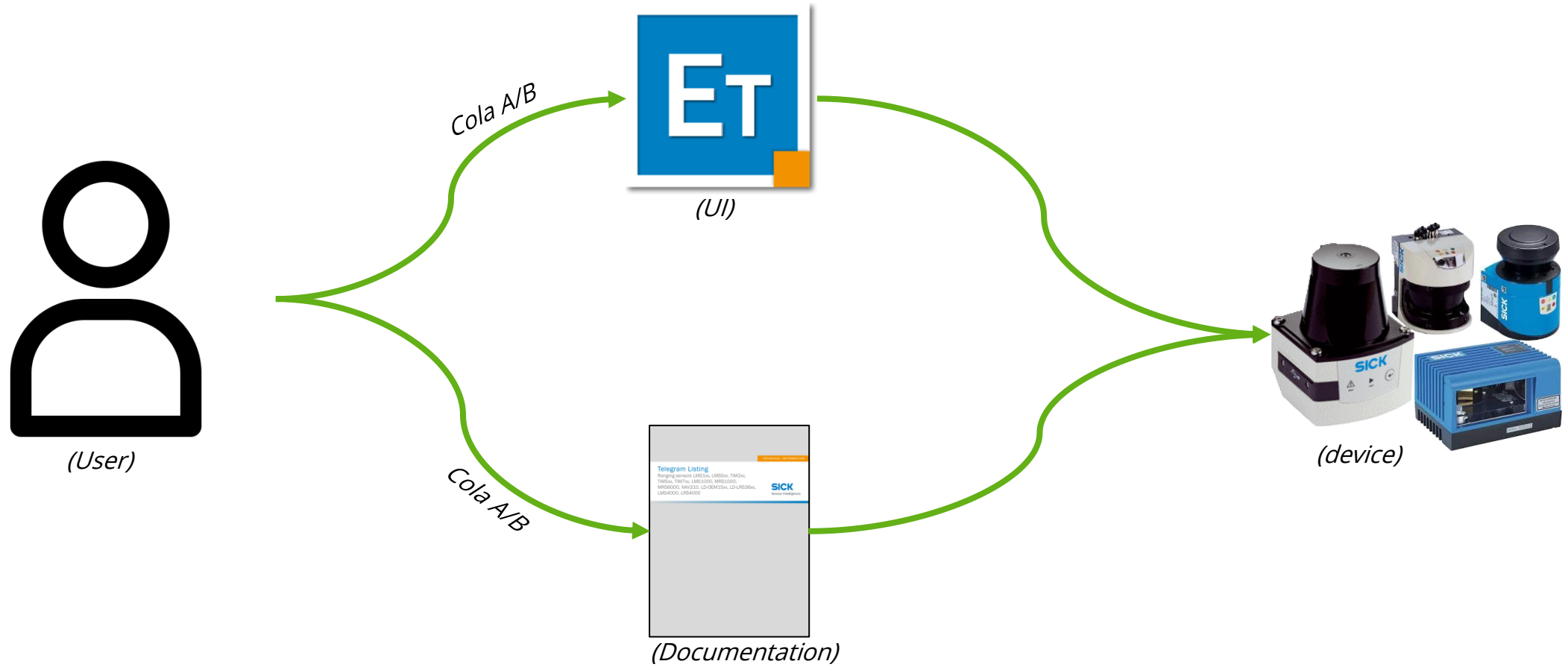
1. Where do we come from (Cola interface)?
2. What is HTTP and REST?
3. What is OpenAPI and which benefits does it bring to the table?

# Given interfaces with (new) devices



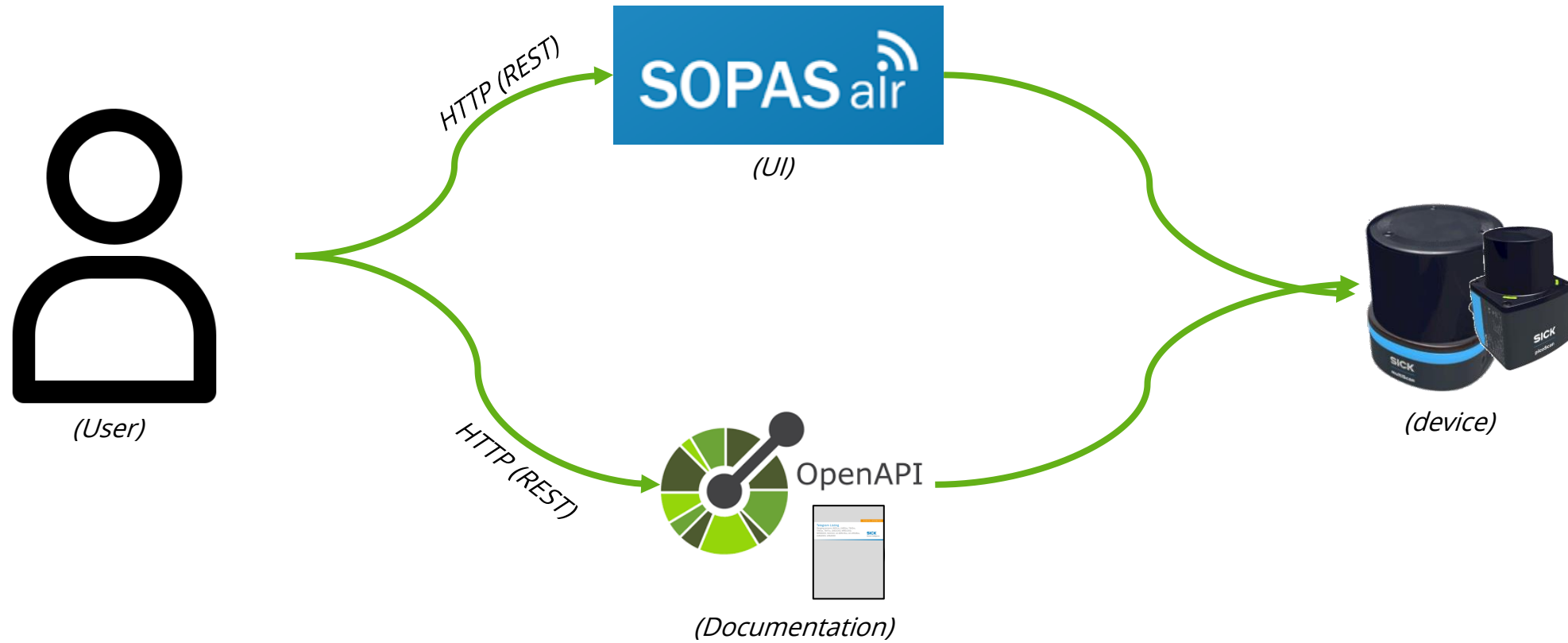
# UI and Telegramm UX

Cola A/B



# UI and Telegramm UX

HTTP (REST)



# Dry theory ☹️

## HTTP (Hypertext Transfer Protocol):

- HTTP is a protocol that server and clients use to communicate on the internet.
- It's the foundation for transferring data on the web.
- About 30 years old



*(**cooking** can be chopping, frying, and baking..)*

## REST (Representational State Transfer):

- REST is a set of rules for how to build and structure web services.
- REST is like a recipe for creating web applications that can talk to each other.
- About 25 years old



*(**one recipe** explains which methods & ingredients need to be used to cook a meal)*

## OpenAPI (formally known as swagger):

- OpenAPI is a document that explains how a web service works.
- OpenAPI makes it easier for developers to understand and work with web services.
- About 10 years old



*(**cookbooks** guide you through recipes to mastering cooking)*

They all work together to make the web function smoothly and allow different software systems to communicate effectively.

# OpenAPI

How does it look like?

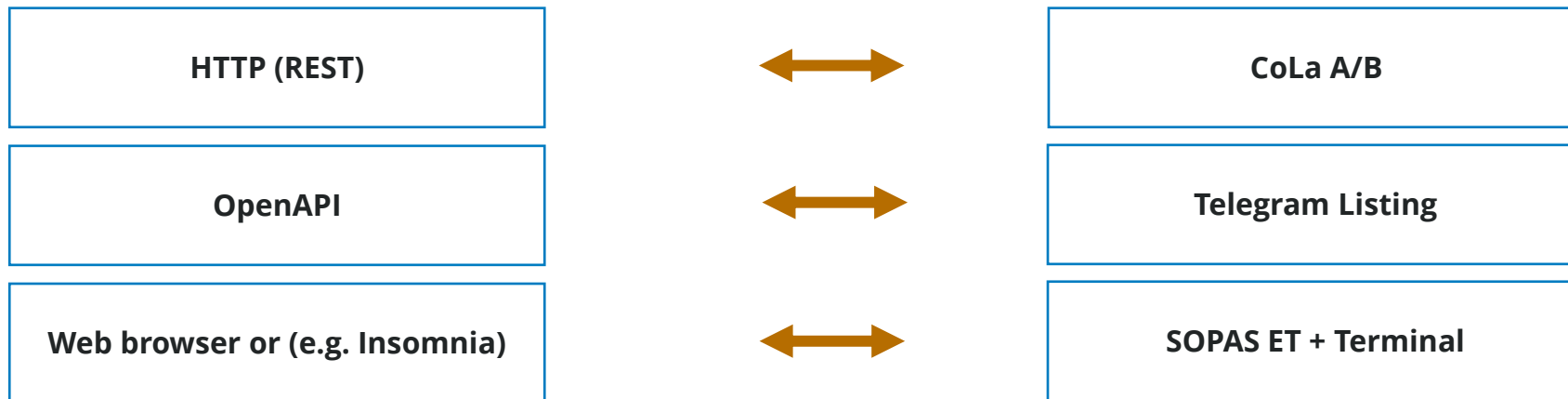
YAML file representation (like a text file)

UI representation

The image shows a side-by-side comparison of an OpenAPI specification. On the left is the raw YAML text, and on the right is a graphical user interface (UI) for the same specification. The UI is divided into several sections, each with a label pointing to it:

- Variable / Method:** Points to the endpoint `GET /LFPmovingAveragingFilter`.
- Description:** Points to the detailed text describing the filter's function: "The filter assigns the moving average of the distances of the previous 'AverageDepth' scans (including the current scan) to the distance channel of the current scan. Points with zero distance are treated as missing values and are not taken into account for averaging. If all distances considered for averaging have the value 0, the result will be zero as well. The filter returns 'nil' until the minimum required number of scans ('AverageDepth') have been received. In contrast to the Scan.MeanFilter no data reduction is performed i.e. the scan input rate and the scan output rate are identical. The input scans are modified in place, therefore use cloning to operate on the original scans outside the filter while the filter is running. For multilayer data, each layer is filtered separately."
- Try Out Button:** Points to the blue `Execute` button.
- Example Code:** Points to the `curl` command: `curl -X 'GET' 'http://192.168.0.1/api/LFPmovingAveragingFilter' -H 'accept: application/json'`
- Example URL:** Points to the request URL: `http://192.168.0.1/api/LFPmovingAveragingFilter`
- Example Response Body:** Points to the JSON response: `{ "header": { "status": 0`

# What to compare?



# OpenAPI at Dynamic Ranging

## Advantages and Information

### **Advantages**

- Open API
  - Easy to read & use
  - Standardized specification more than 10 years old
  - Used at SICK already: FTMG, TDC-E, SIG350
  - Has a rich open-source toolset (Visualization, converters, code generation, .. <https://openapi.tools/> )
  - Includes examples → Documentation offers a tryout button
  - Good solution for future security standards
- HTTP
  - Is a well known, often used, non-proprietary protocol and more secure than Cola as used with Challenge and response method.

### **Additional Information**

- Does not publish the measurement data protocol (compact, MSGPACK)
- Due to the use of HTTP, there are no events anymore
- Due to cyber security and the used Challenge and Response method write / POST methods are not so simple anymore → Python example in development
- It is used by multiple customers