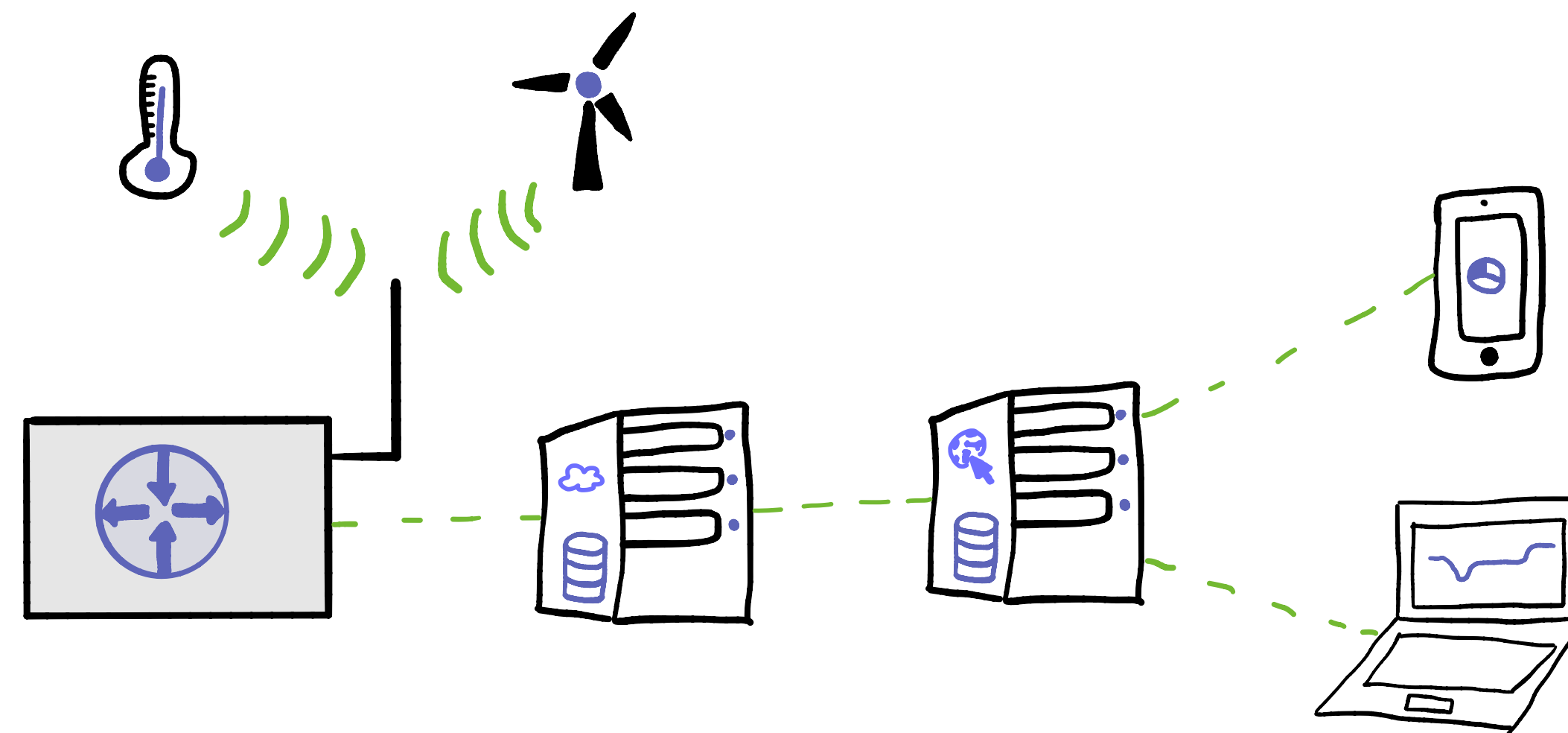


Internet of Things mit LoRaWAN

Workshop am SFZ Ochsenhausen



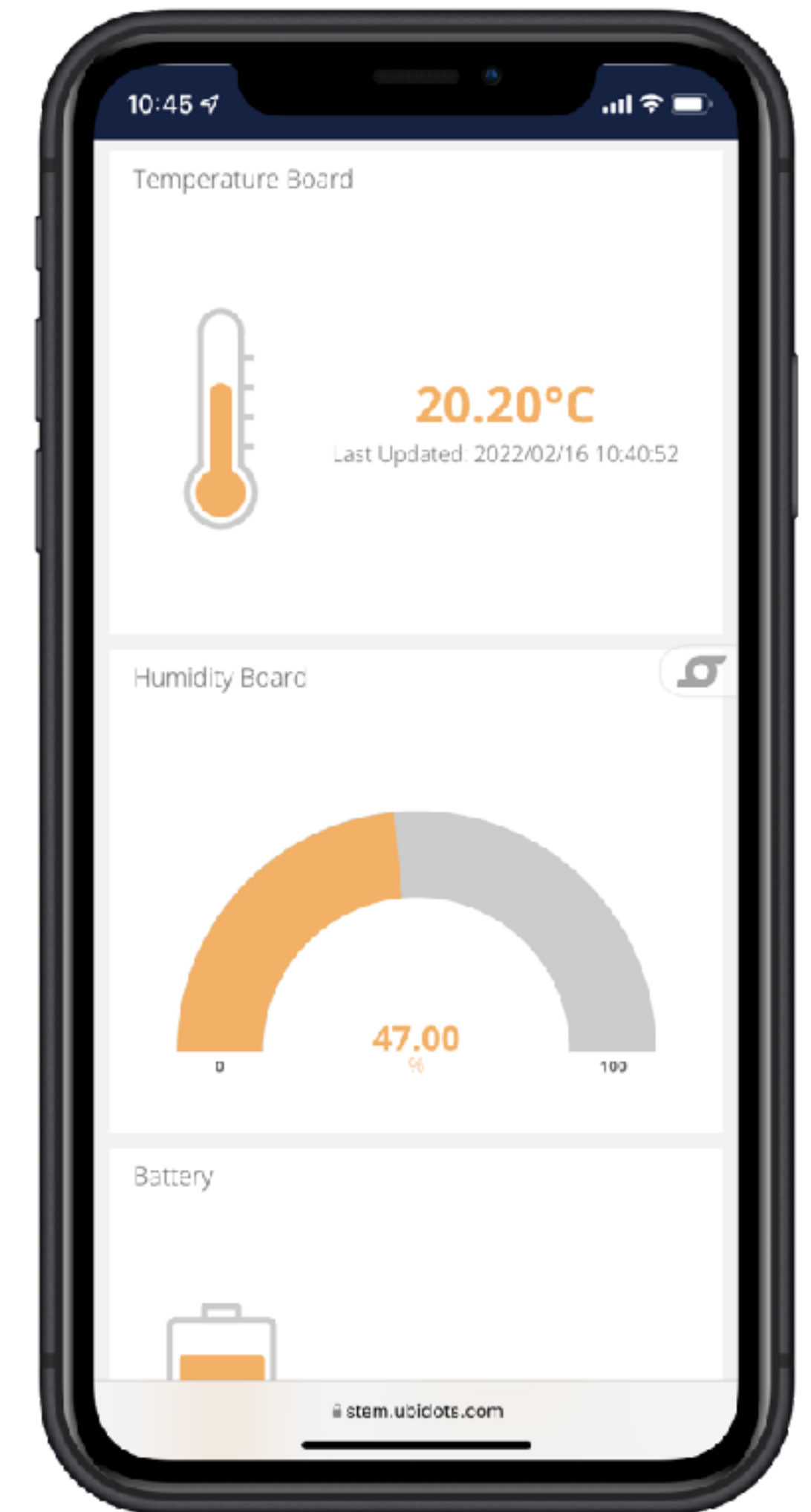
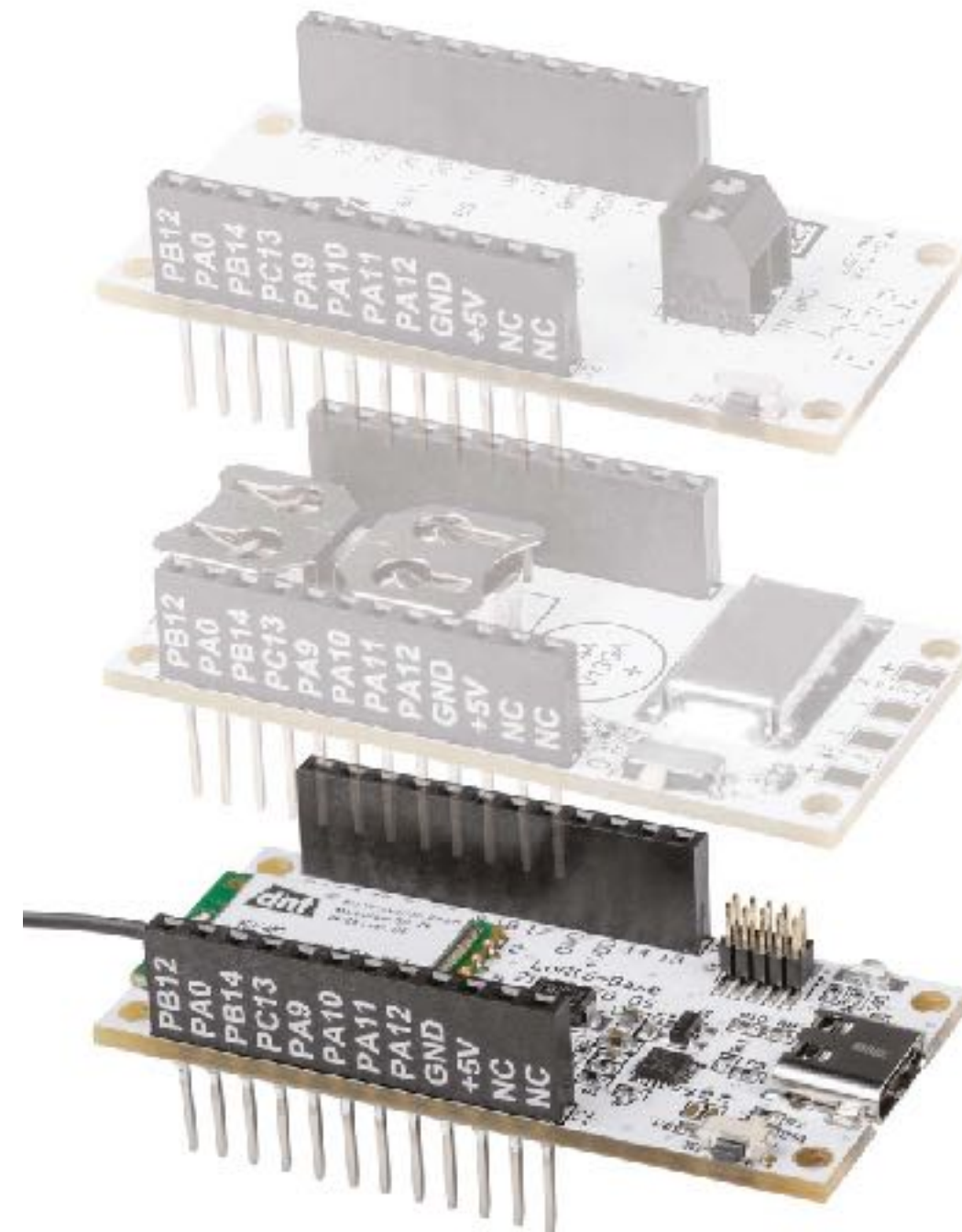
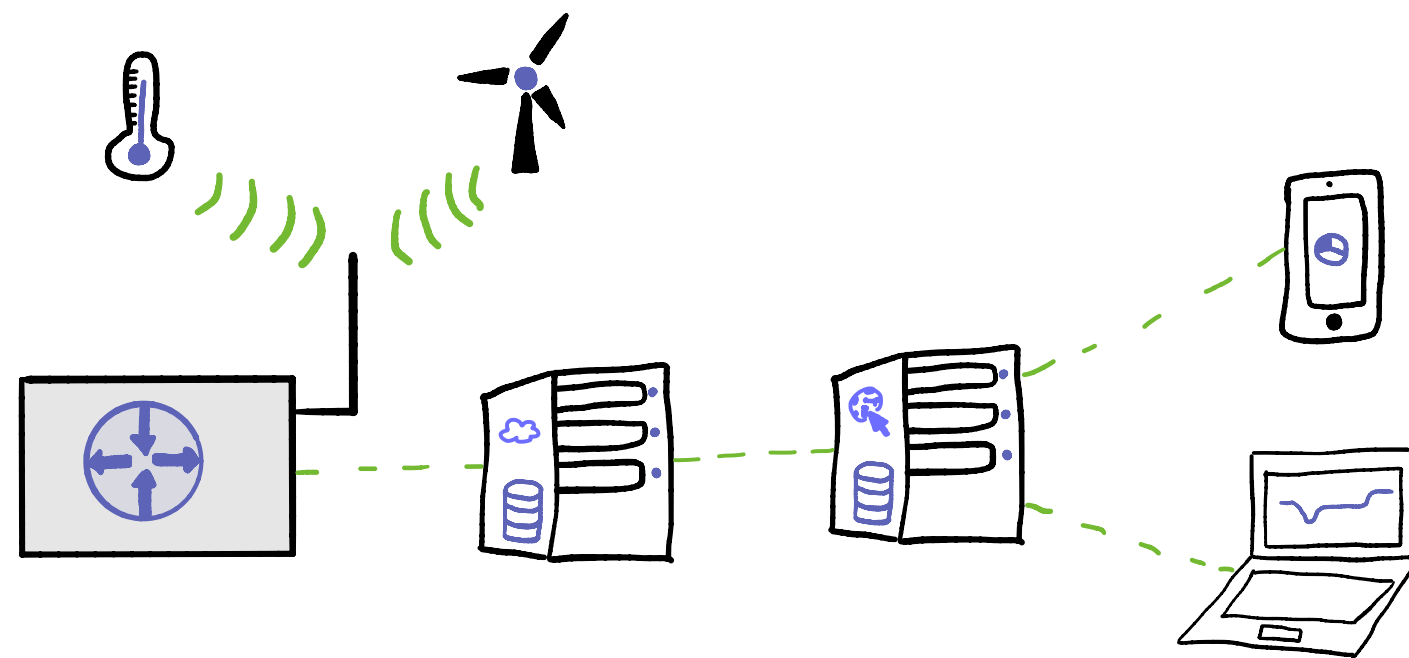
Alexander Knaak, Benno Hölz

Das Internet der Dinge

(Internet of Things, IOT)

„Das **Internet der Dinge** ist ein Sammelbegriff für Technologien einer globalen Infrastruktur der Informationsgesellschaften, die es ermöglicht, **physische und virtuelle Objekte miteinander zu vernetzen** und sie durch Informations- und Kommunikationstechniken **zusammenarbeiten** zu lassen.“

Was machen wir?



1. Was ist LoRaWAN?

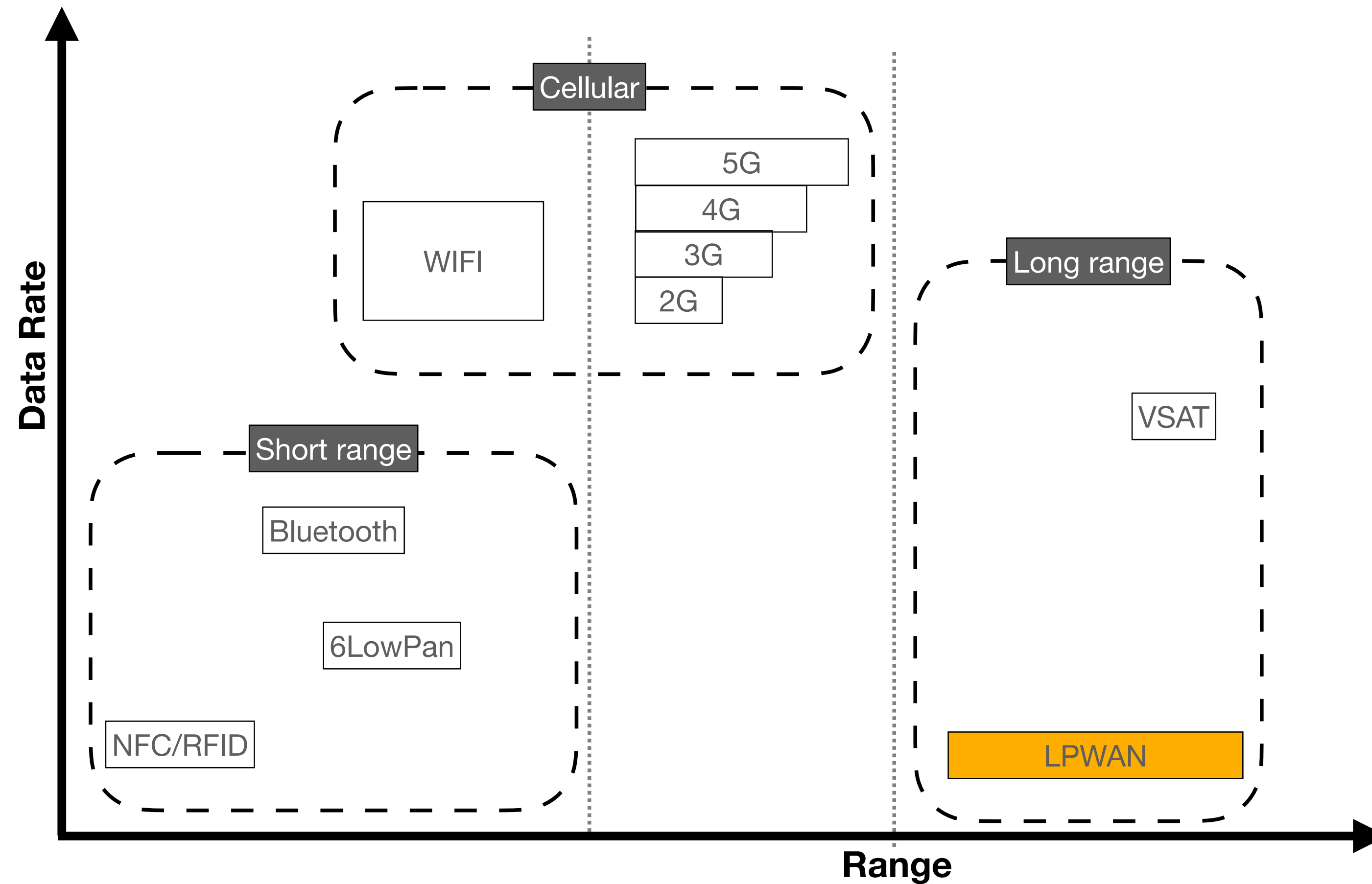
Was ist LoRaWAN?

Definition

- Funkstandard:
 - Kleine Datenmengen
 - Große Entfernungen (**LongRange**)
 - Niedriger Energiebedarf
- oft verwendet für Sensoren
- **LoRaWAN**: Netzwerk aus Sensoren/Aktoren (WAN = **W**ide **A**rea **N**etwork)

Was ist LoRaWAN?

Übertragungstechnik - Frequenzbänder



Was ist LoRaWAN?

Übertragungstechnik - Technische Daten

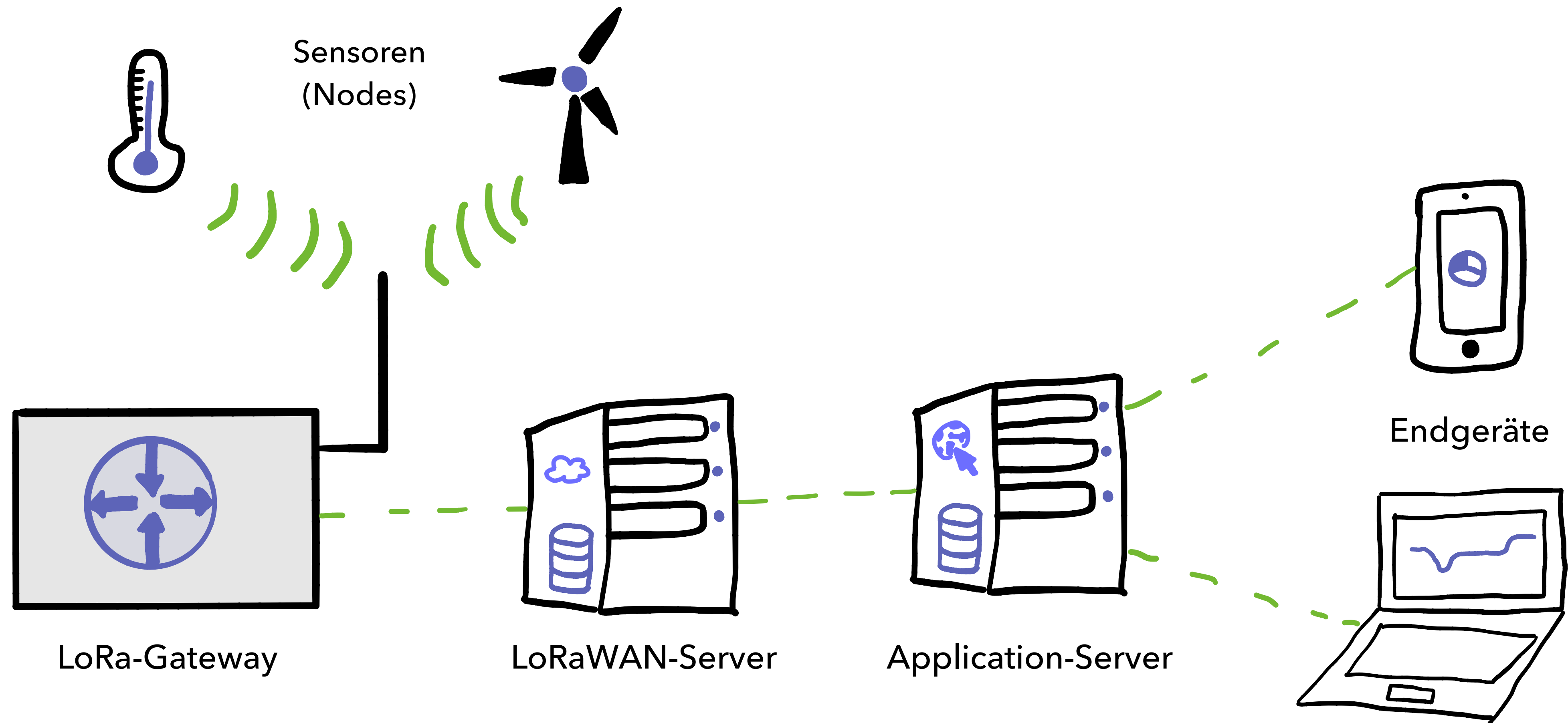
	Europa
Frequenzband	867 - 869 MHz
Kanäle	10
Sendeleistung	+14 dBm $\triangleq$ 25 mW
Spreading Factor (SF)	7-12
Datenrate	250 - 11000 bit/s

ACHTUNG:

Sendezeit ist in Europa begrenzt:
Maximal 864 Sekunden/Tag

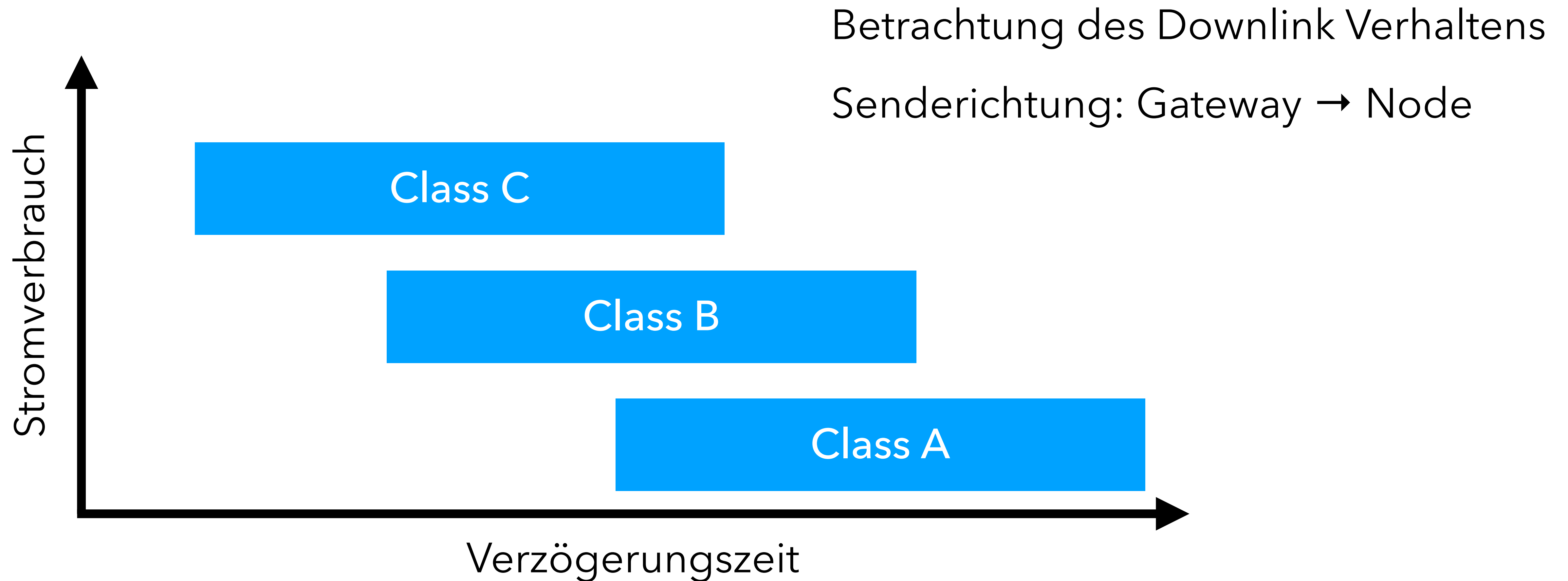
Was ist LoRaWAN?

Das Netzwerk



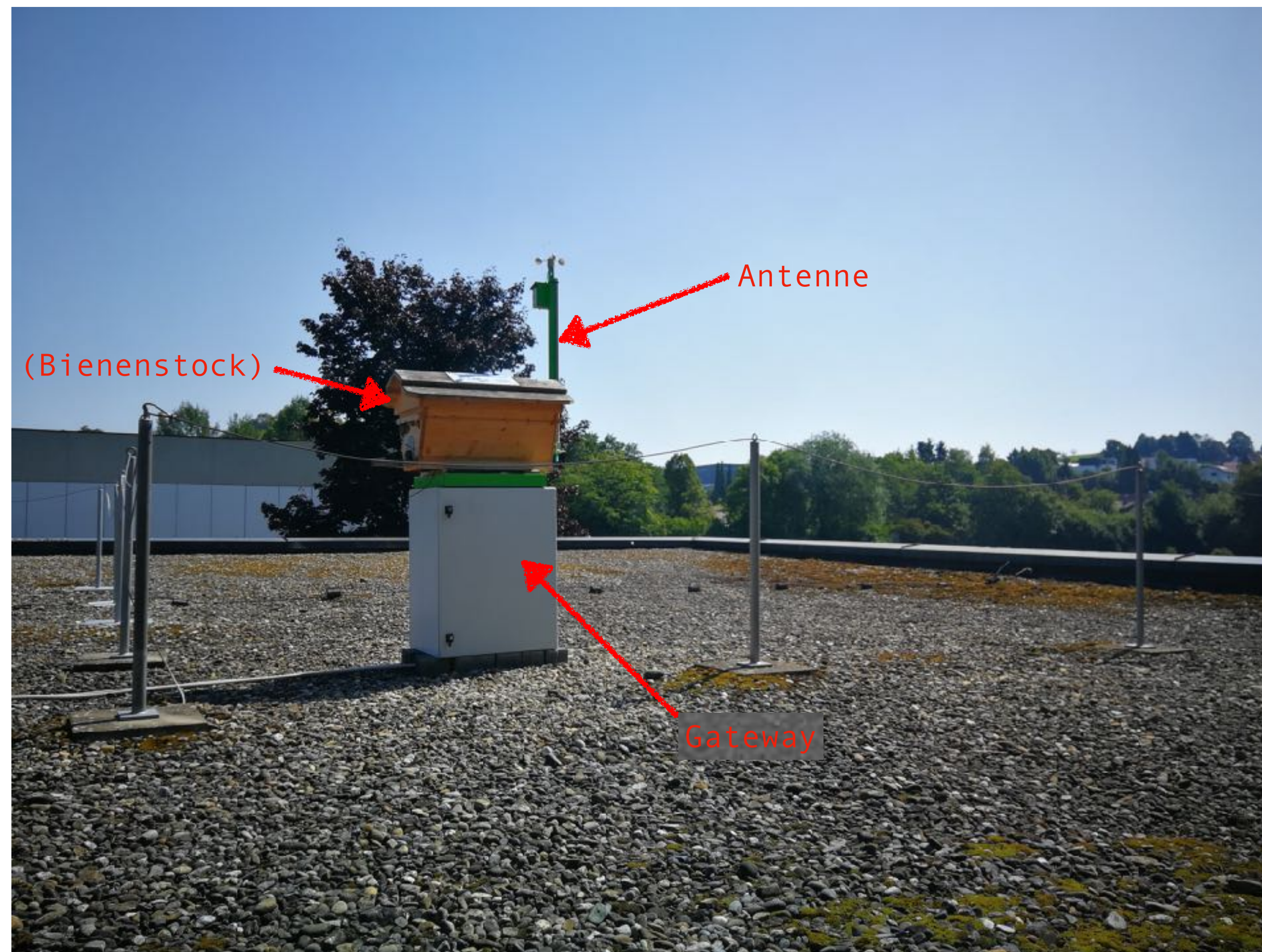
Was ist LoRaWAN?

Geräteklassen



Was ist LoRaWAN?

Gateway am SFZ Ochsenhausen

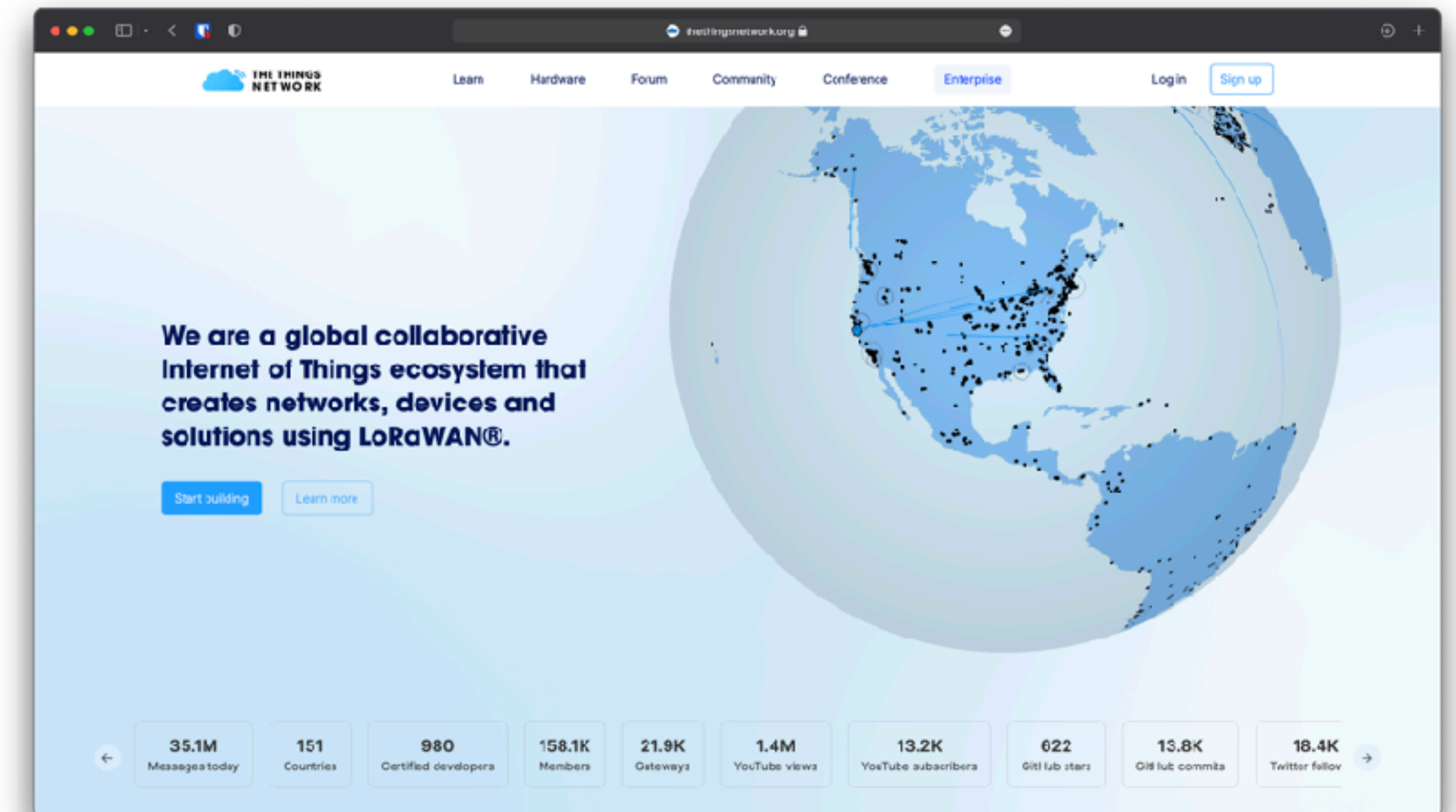


2. The Things Network

The Things Network (TTN)

Ein IOT-Netzwerk

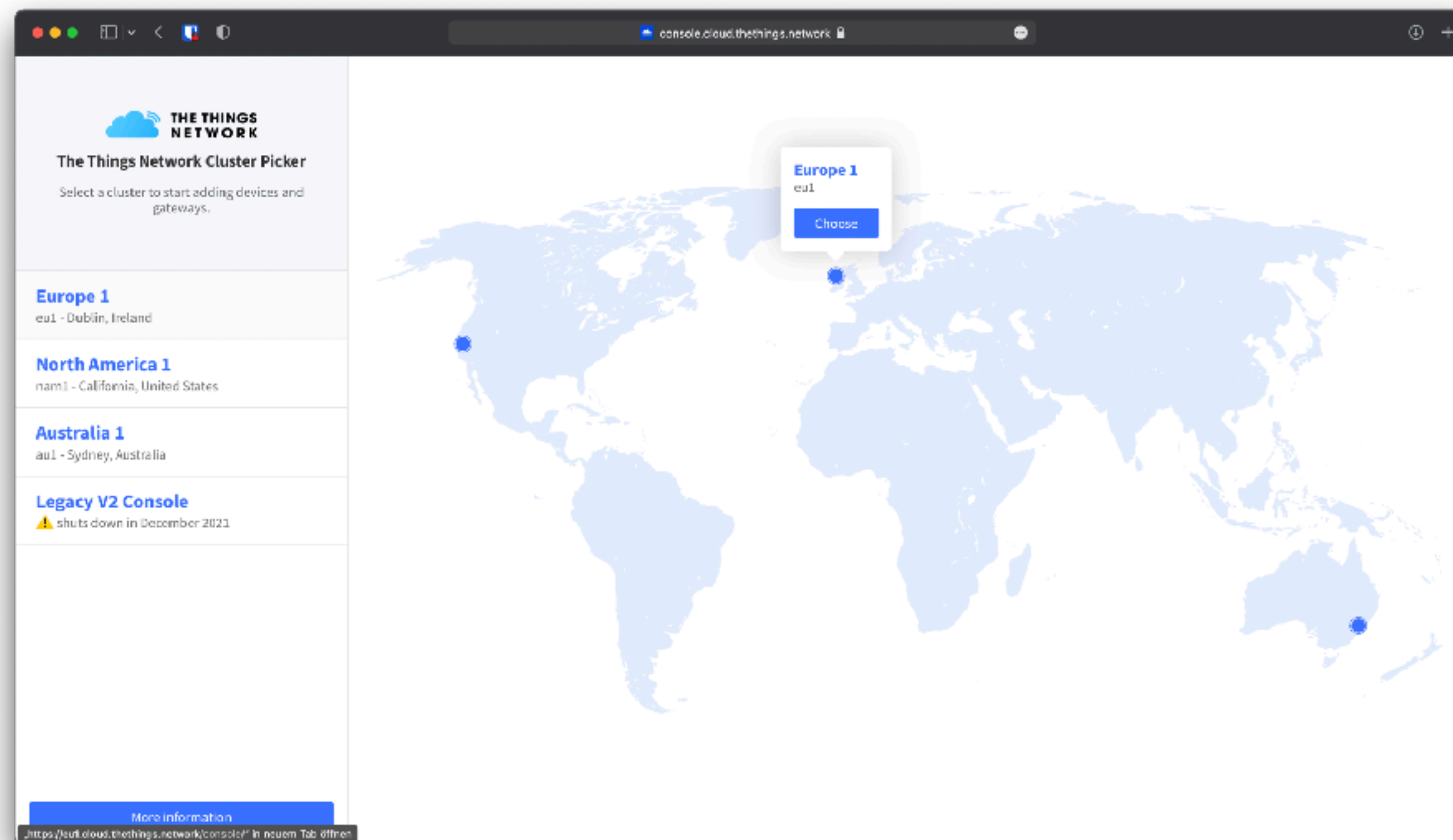
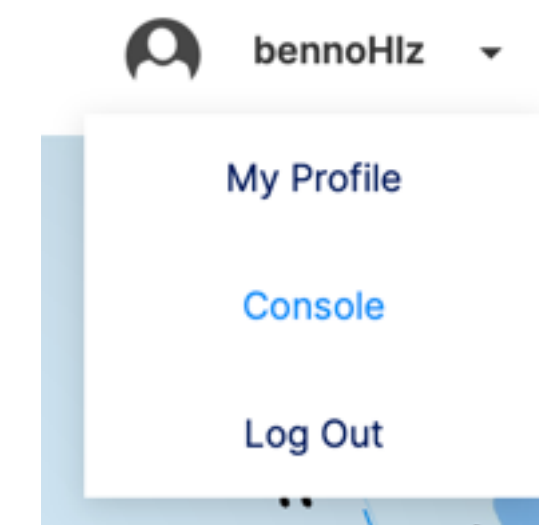
- IOT-Netzwerk für LoRaWAN-Sensoren
- Aktuelle Version: TheThingsStack v3
- Account erstellen unter:
<https://www.thethingsnetwork.org>



The Things Network (TTN)

Die Console

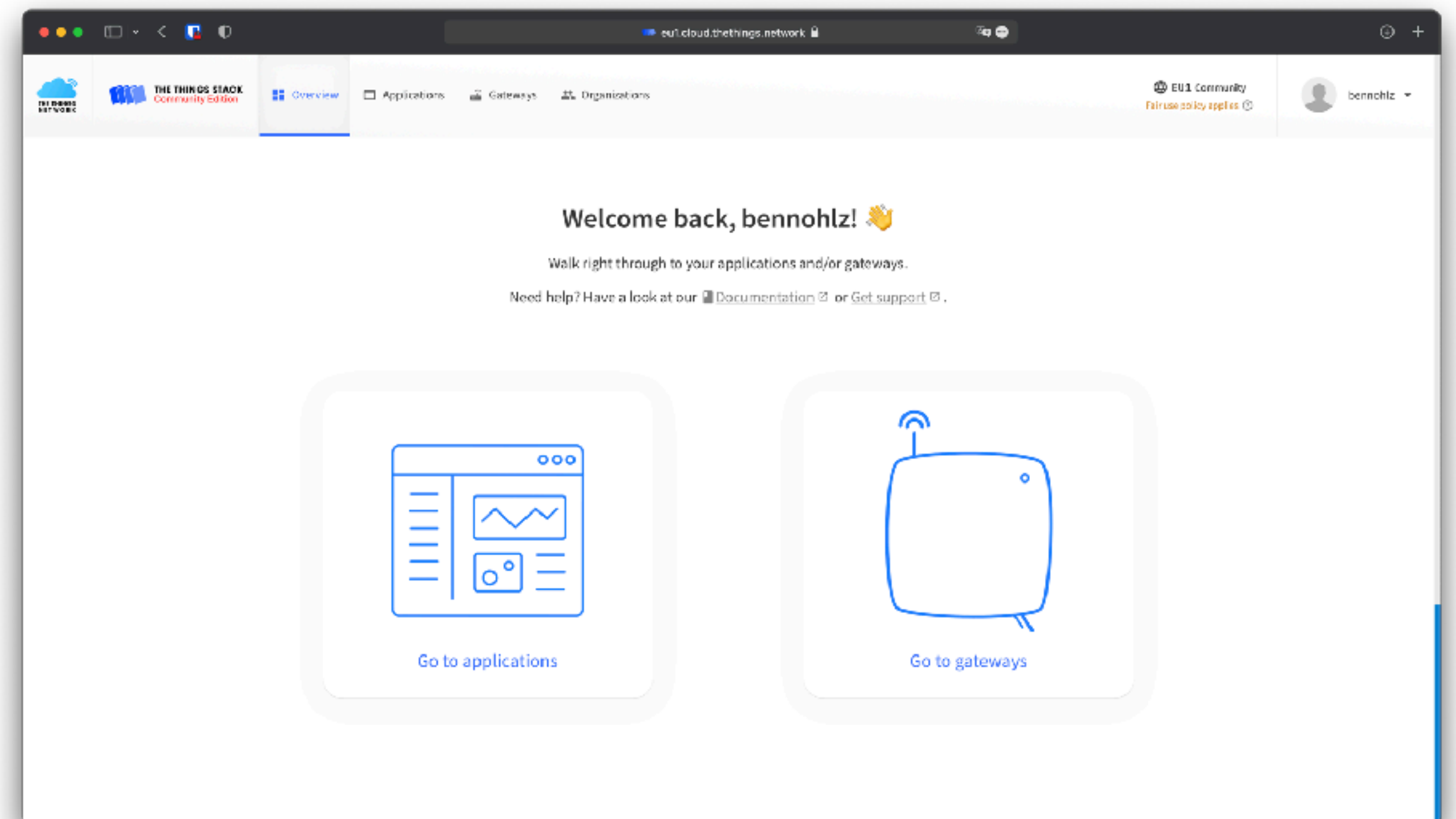
- Erreichbar über **Benutzername > Console**
- Auswahl der Region **Europe 1**



The Things Network (TTN)

Die Console

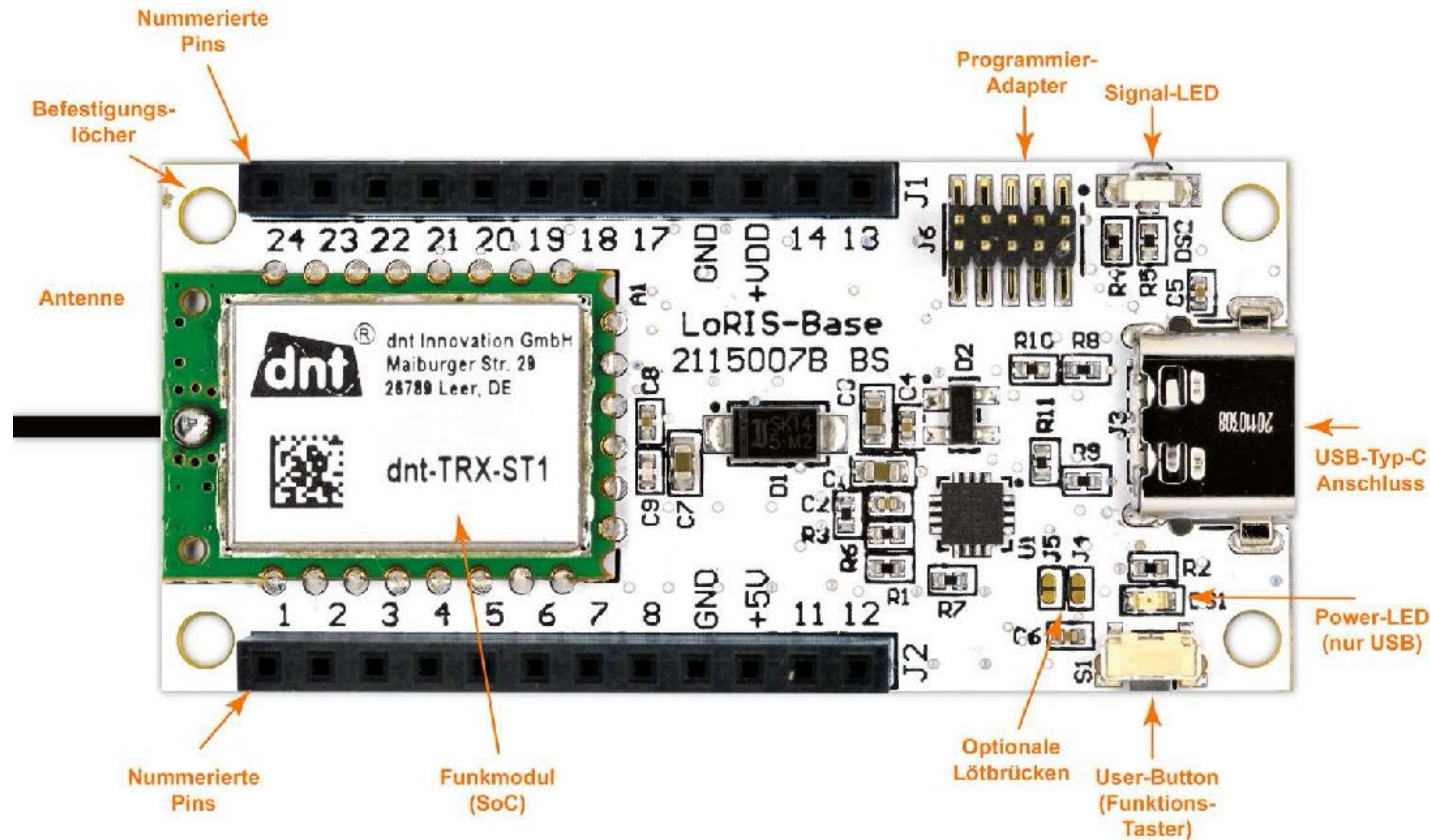
- Zwei Bereiche: Applications und Gateways
→ Für uns wichtig: **Applications**



3. Die LoRIS-Base Entwicklungsplattform

Die LoRIS-Base Entwicklungsplattform

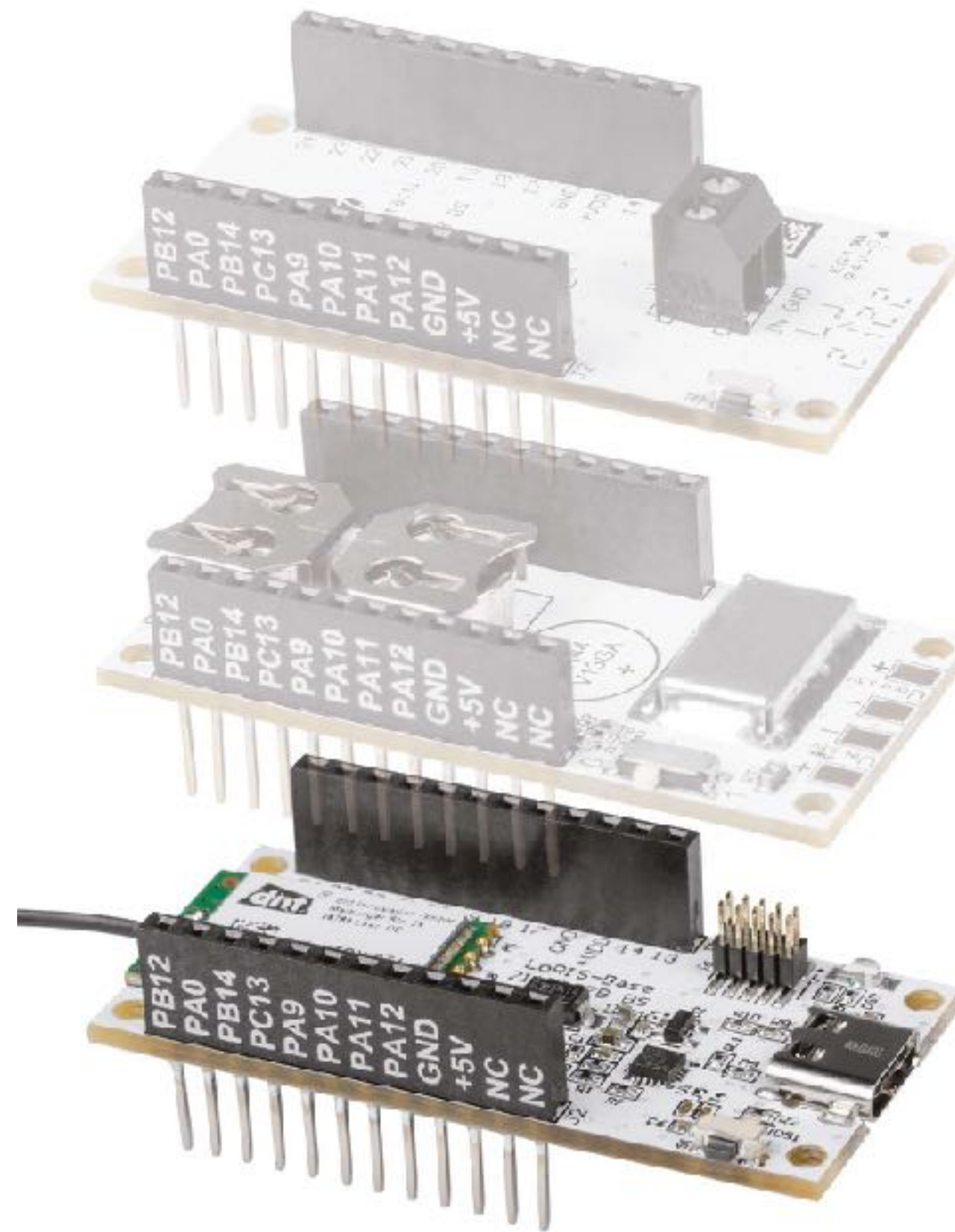
Das Board



Quelle: <https://de.elv.com/lorawan>

Die LoRS-Base Entwicklungsplattform

Verschiedene Erweiterungsmodule



Quelle: <https://de.elv.com/lorawan>

4. Board in TTN einbinden

Board in TTN einbinden

Aktivierungsmethoden

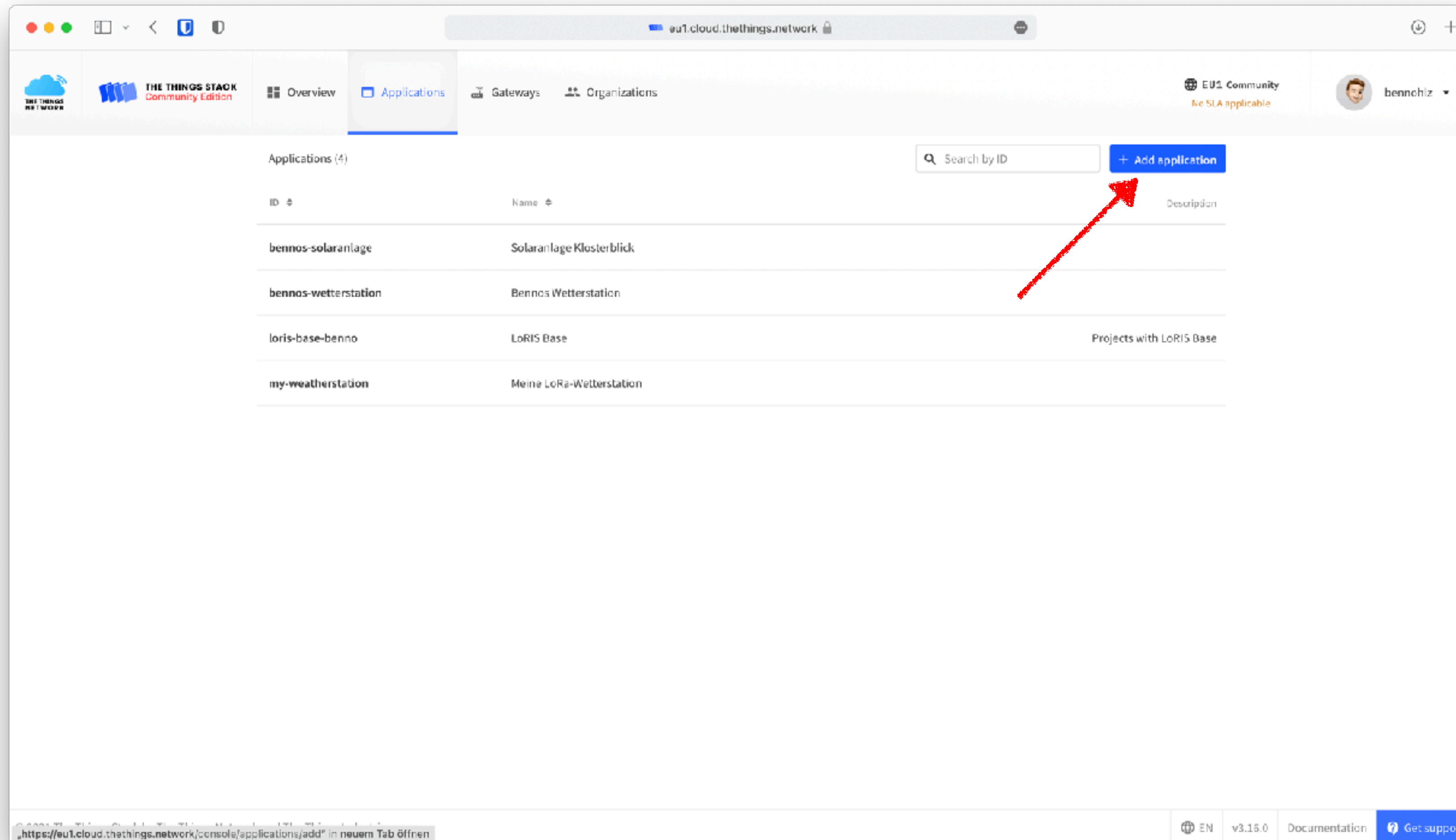
2 Möglichkeiten, ein Device zu aktivieren:

- **OTAA** (**O**ver-**T**he-**A**ir **A**ctivation)
- **ABP** (**A**ctivation **B**y **P**ersonalization)

—> Wir verwenden OTAA

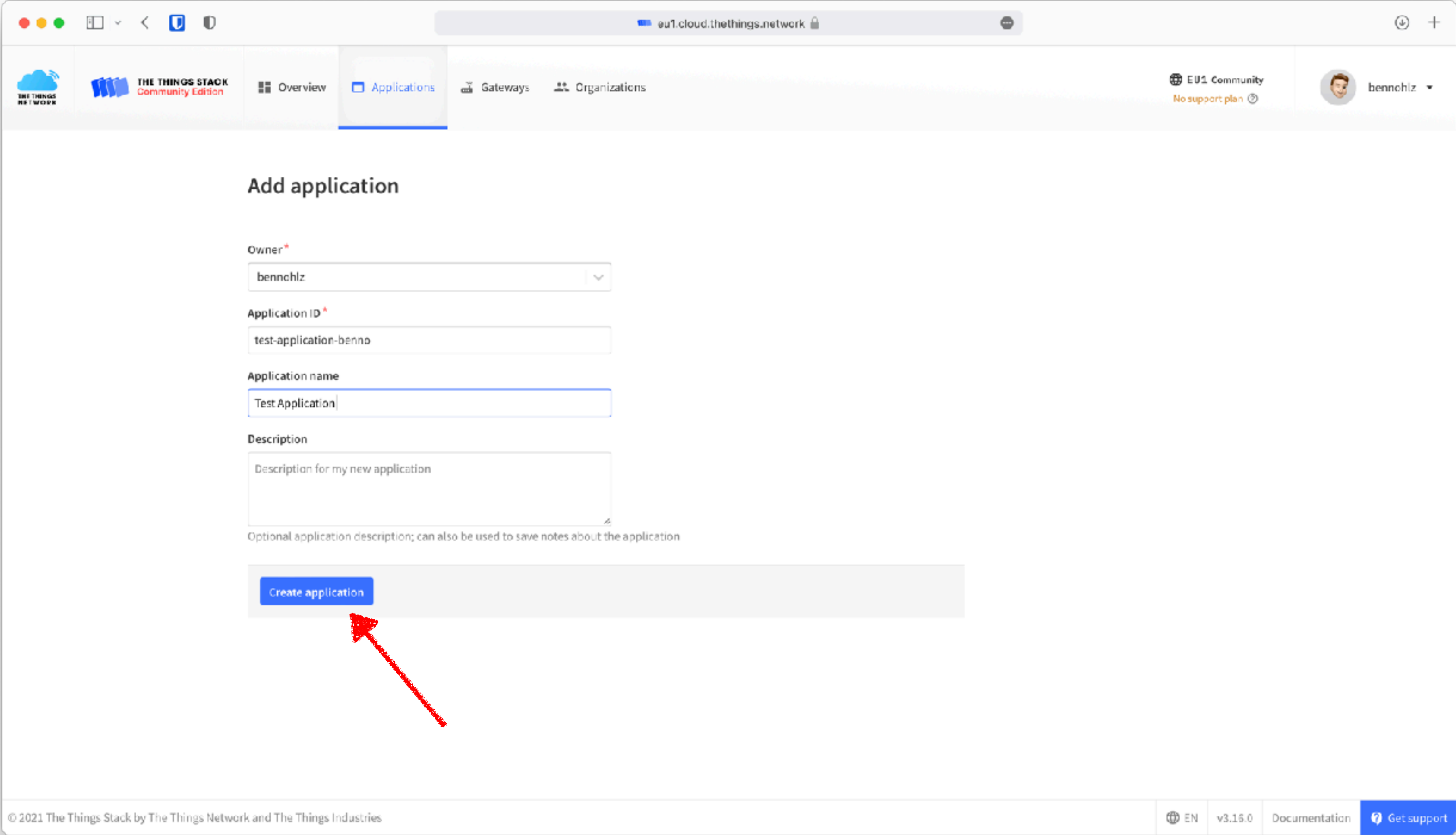
Board in TTN einbinden

Application erstellen



Board in TTN einbinden

Application erstellen



The screenshot shows the TTN (The Things Network) web interface. The browser address bar displays `eu1.cloud.thethings.network`. The navigation bar includes the TTN logo, 'THE THINGS STACK Community Edition', and tabs for 'Overview', 'Applications' (which is selected), 'Gateways', and 'Organizations'. On the right, it shows 'EU1 Community', 'No support plan', and a user profile for 'bennchiz'.

The main content area is titled 'Add application'. It contains the following form fields:

- Owner***: A dropdown menu with 'bennchiz' selected.
- Application ID***: A text input field containing 'test-application-benno'.
- Application name**: A text input field containing 'Test Application'.
- Description**: A text area containing 'Description for my new application'.

Below the description field, there is a note: 'Optional application description; can also be used to save notes about the application'.

A blue button labeled 'Create application' is located at the bottom of the form. A red arrow points to this button.

The footer contains the copyright notice '© 2021 The Things Stack by The Things Network and The Things Industries', language and version information 'EN v3.15.0', a link to 'Documentation', and a 'Get support' button.

Board in TTN einbinden

Device anlegen

The screenshot shows the TTN (The Things Network) web interface. The browser address bar displays `eu1.cloud.thethings.network`. The top navigation bar includes links for Overview, Applications (selected), Gateways, and Organizations. The user profile 'bennchiz' is visible in the top right corner.

The left sidebar contains the following menu items: Test Application, Overview (selected), End devices, Live data, Payload formatters, Integrations, Collaborators, API keys, and General settings.

The main content area is titled 'Test Application' with ID 'test-application-benno'. It shows 'No recent activity' and statistics: 0 End devices, 1 Collaborator, and 0 API keys.

Under 'General information', the following details are listed:

- Application ID: test-application-benno
- Created at: Nov 19, 2021 15:35:42
- Last updated at: Nov 19, 2021 15:35:42

The 'Live data' section shows a message: 'Waiting for events from test-application-benno...'. A link 'See all activity' is available.

At the bottom, there is a section for 'End devices (0)' with a search bar 'Search by ID', buttons for 'Import end devices' and 'Add end device' (highlighted with a red arrow), and a table with columns: ID, Name, DevEUI, JoinEUI, and Last activity. The table currently shows 'No items found'.

The footer contains copyright information: '© 2021 The Things Stack by The Things Network and The Things Industries', language 'EN', version 'v3.15.0', a link to 'Documentation', and a 'Get support' button.

Board in TTN einbinden

Device anlegen

The screenshot shows the 'Register end device' page in the TTN console. The 'Manually' tab is selected. The form includes the following fields and controls:

- Frequency plan**: Europe 863-870 MHz (SF9 for RX2 - recommended)
- LoRaWAN version**: MAC V1.0.3
- Regional Parameters version**: PHY V1.0.3 REV A
- Show advanced activation, LoRaWAN class and cluster settings**: expanded
- DevEUI**: [redacted] (with 'Generate' button)
- AppEUI**: [redacted] (with 'Generate' button)
- AppKey**: [redacted] (with 'Generate' button)
- End device ID**: my-new-device (with note: 'This value is automatically prefilled using the DevEUI')
- After registration**: ☒ View registered end device, ☐ Register another end device of this type
- Register end device**: blue button at the bottom

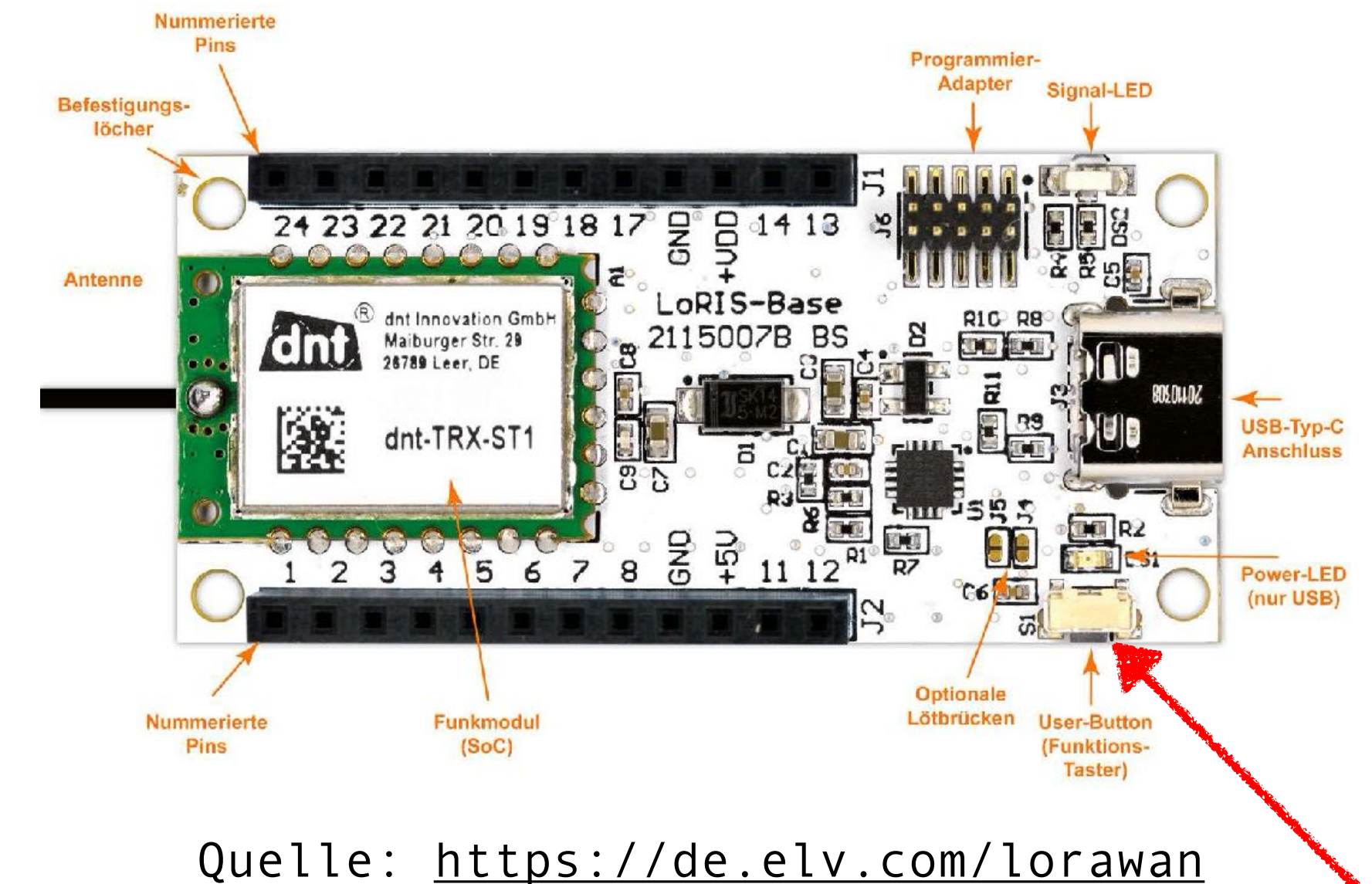
Red arrows point to the 'Manually' tab, the 'Frequency plan', 'LoRaWAN version', 'DevEUI', 'AppEUI', 'AppKey', and the 'Register end device' button.

Daten vom Sticker
des Boards

Board in TTN einbinden

Test-Verbindung herstellen

- Board an die Stromversorgung anschließen
- LED blinkt **orange**: Verbindungsaufbau
- LED leuchtet **grün** auf: Verbindung erfolgreich
- LED leuchtet **rot** auf: Verbindungsfehler
- Versenden einer Testnachricht durch Drücken des User-Buttons
- Nachricht erscheint unter **Live Data**:



↑ 33 ↓ 6 • Last activity 28 seconds ago

Overview **Live data** Messaging Location Payload formatters Claiming General settings

Time	Type	Data preview
↑ 15:47:17	Forward uplink data message	Payload: { Supply_Voltage: 3031, TH_Sensor_Humidity: "44", TH_Sensor_Temperature: "29.9", TX_Reason: "User Button", Temperature_Sensor: "19.2" }

Verbose stream ☐ Pause Clear

Board in TTN einbinden

Payload-Formatter

The screenshot displays the TTN web interface for configuring a payload formatter. The browser address bar shows `eu1.cloud.thethings.network`. The left sidebar contains navigation links: Test Application, Overview, End devices (selected), Live data, Payload formatters, Integrations, Collaborators, API keys, and General settings. The main content area shows the configuration for end device `eui-70b3d57ed004897b`. The `Payload formatters` tab is active, with sub-tabs for `Uplink` and `Downlink`. The `Uplink` sub-tab is selected, indicated by a red arrow. The `Setup` section includes a `Formatter type` dropdown set to `Javascript` (pointed to by a red arrow) and a `Formatter parameter` text area containing a JavaScript decoder function (pointed to by a red arrow). The `Test` section on the right has a `Byte payload` input field, an `FPort` dropdown set to `1`, and a message: `No test result generated yet`. At the bottom, there is a `Save changes` button (pointed to by a red arrow) and a `Test decoder` button.

eu1.cloud.thethings.network

THE THINGS STACK Community Edition

Overview Applications Gateways Organizations

EU1 Community No SLA applicable

bennchiz

Applications > Test Application > End devices > eui-70b3d57ed004897b > Payload formatters > Uplink

eui-70b3d57ed004897b
ID: eui-70b3d57ed004897b

↑ n/a ↓ n/a • No activity yet

Overview Live data Messaging Location **Payload formatters** Claiming General settings

Uplink Downlink

Setup

Formatter type*
Javascript

Formatter parameter*

```
1 var tx_reason = [ "Timer_Event", "User_Button", "Input_Event" ];
2
3 function Decoder( bytes, port )
4 {
5   var decoded = {};
6
7   if( port === 10 )
8   {
9     if( bytes.length >= 5 )
10    {
11      decoded.TX_Reason = tx_reason[ ( bytes[0] & 0x0F ) ];
12      decoded.Supply_Voltage = ( bytes[3] << 8 ) | bytes[4];
13    }
14    if( bytes[5] === 0x00 )
15    {
```

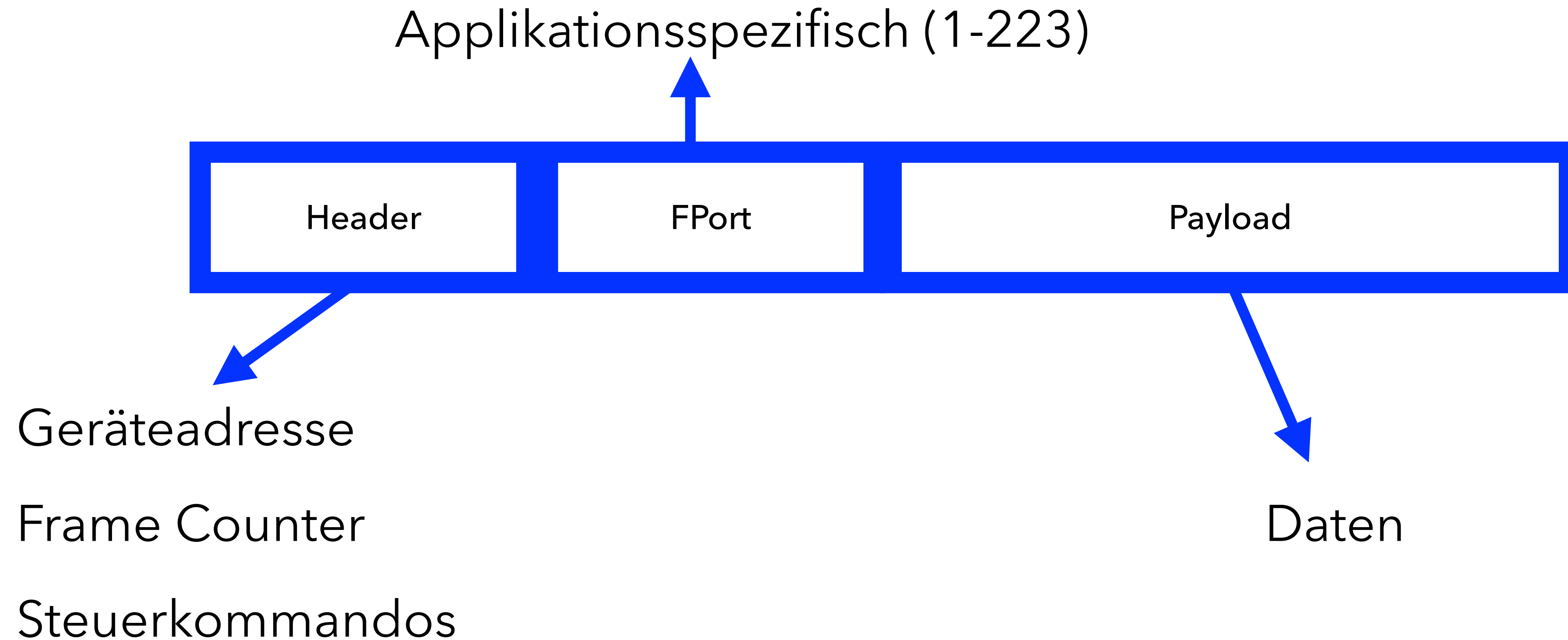
Test

Byte payload FPort
1

No test result generated yet

Save changes Test decoder

Aufbau eines Frames

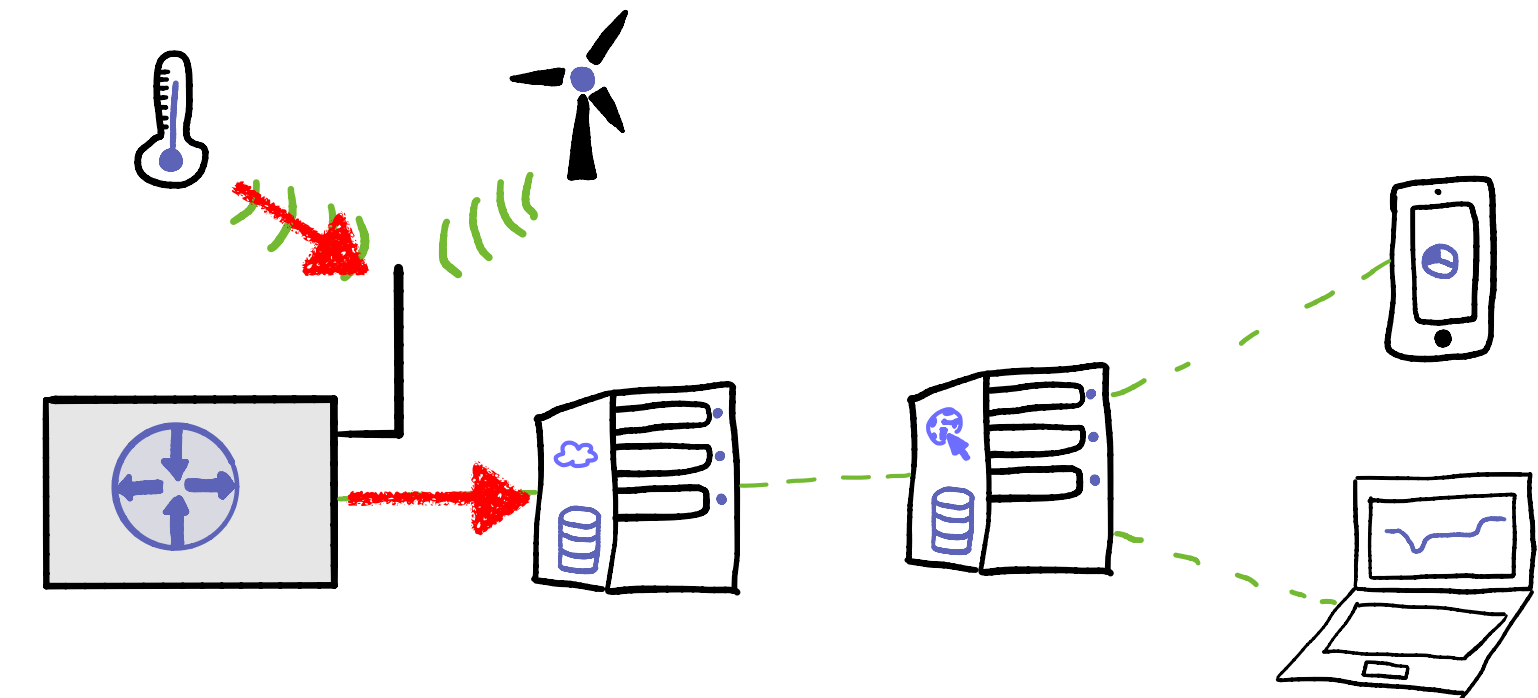


5. Versenden von Nachrichten

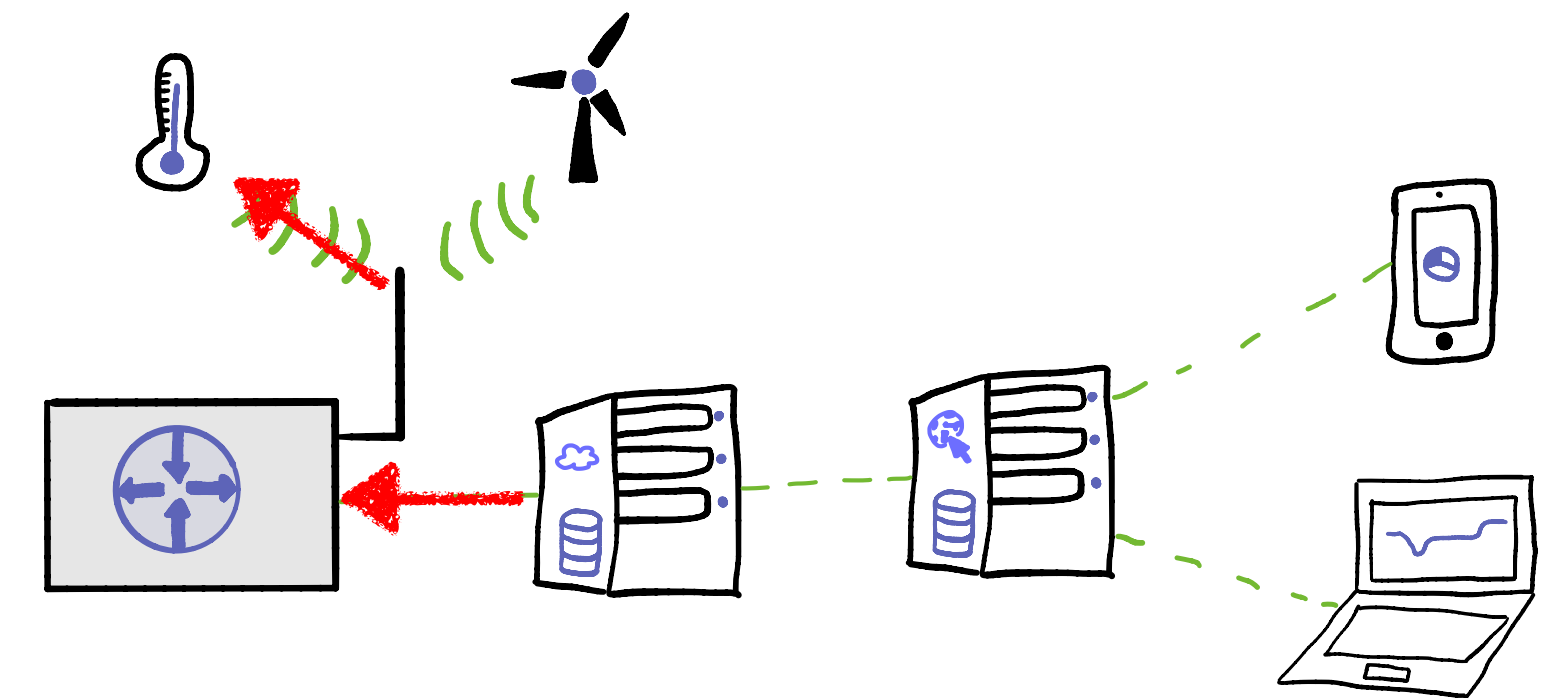
Versenden von Nachrichten

Uplink vs. Downlink

- **Uplink**-Nachricht: Vom Sensor zum Netzwerk

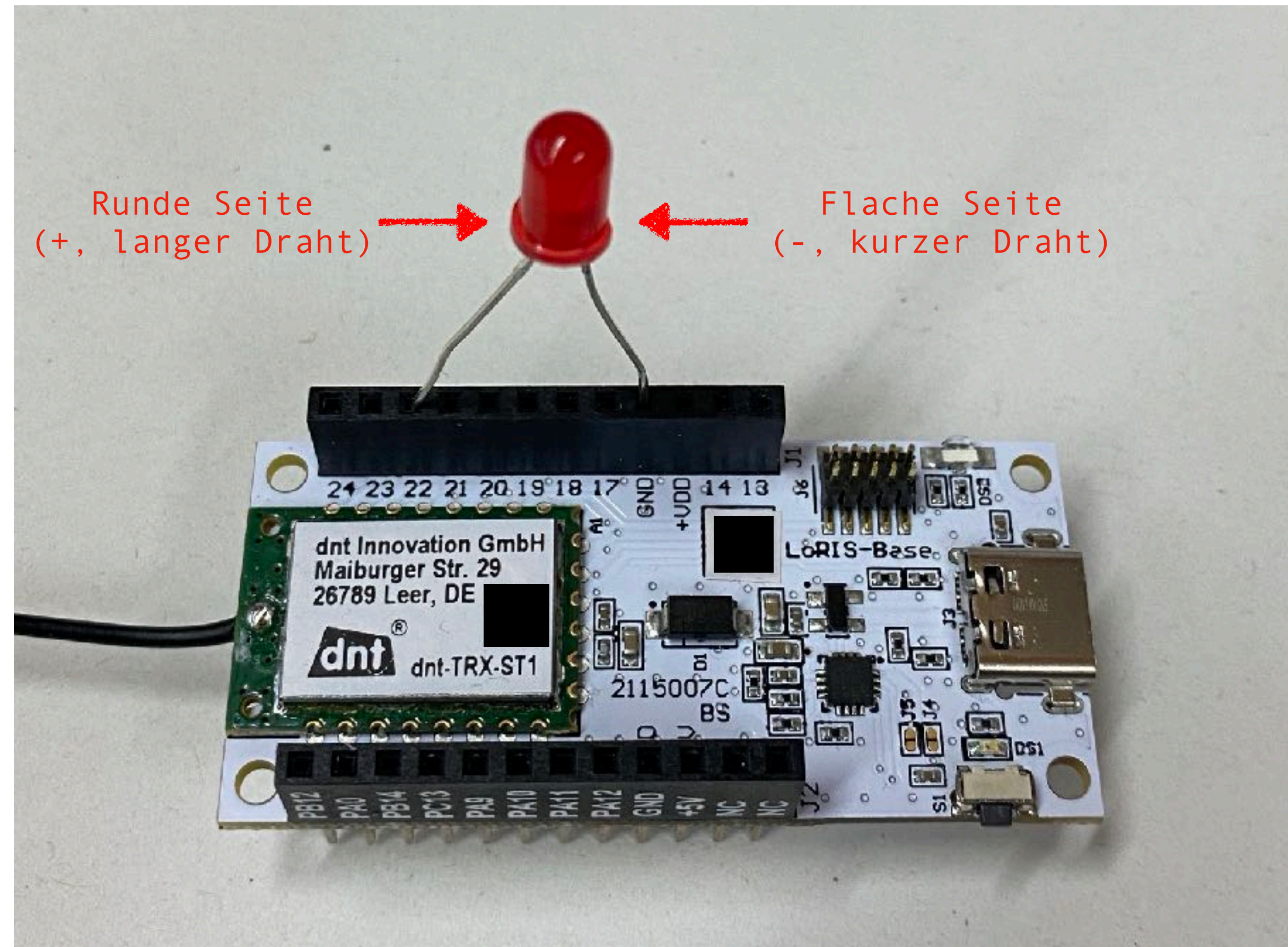


- **Downlink**-Nachricht: Vom Netzwerk zum Sensor
→ Einreihen der Nachricht in Warteschlange
(wird bei nächster Uplink-Nachricht versendet)



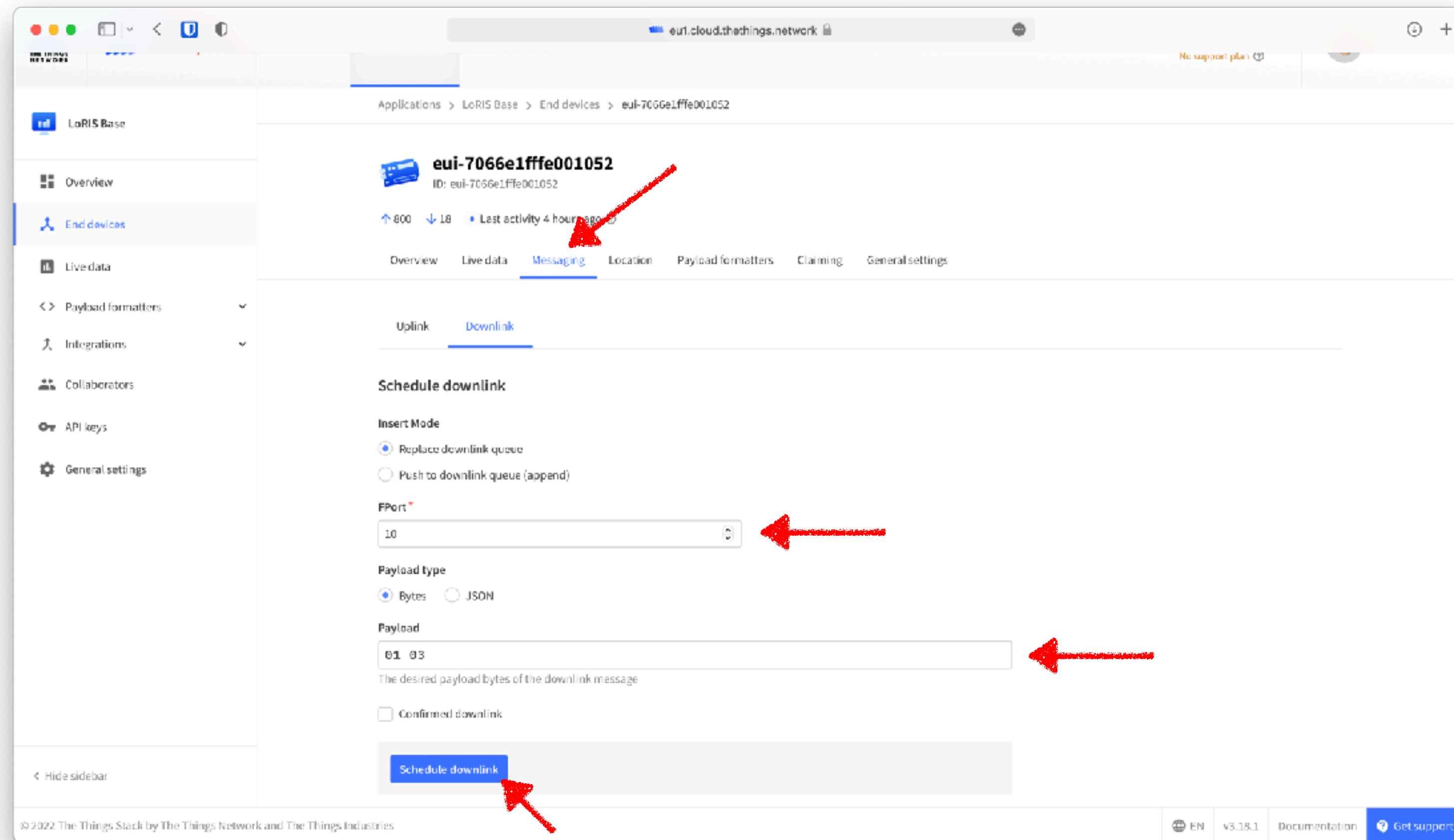
Versenden von Nachrichten

Downlink: LED steuern



Versenden von Nachrichten

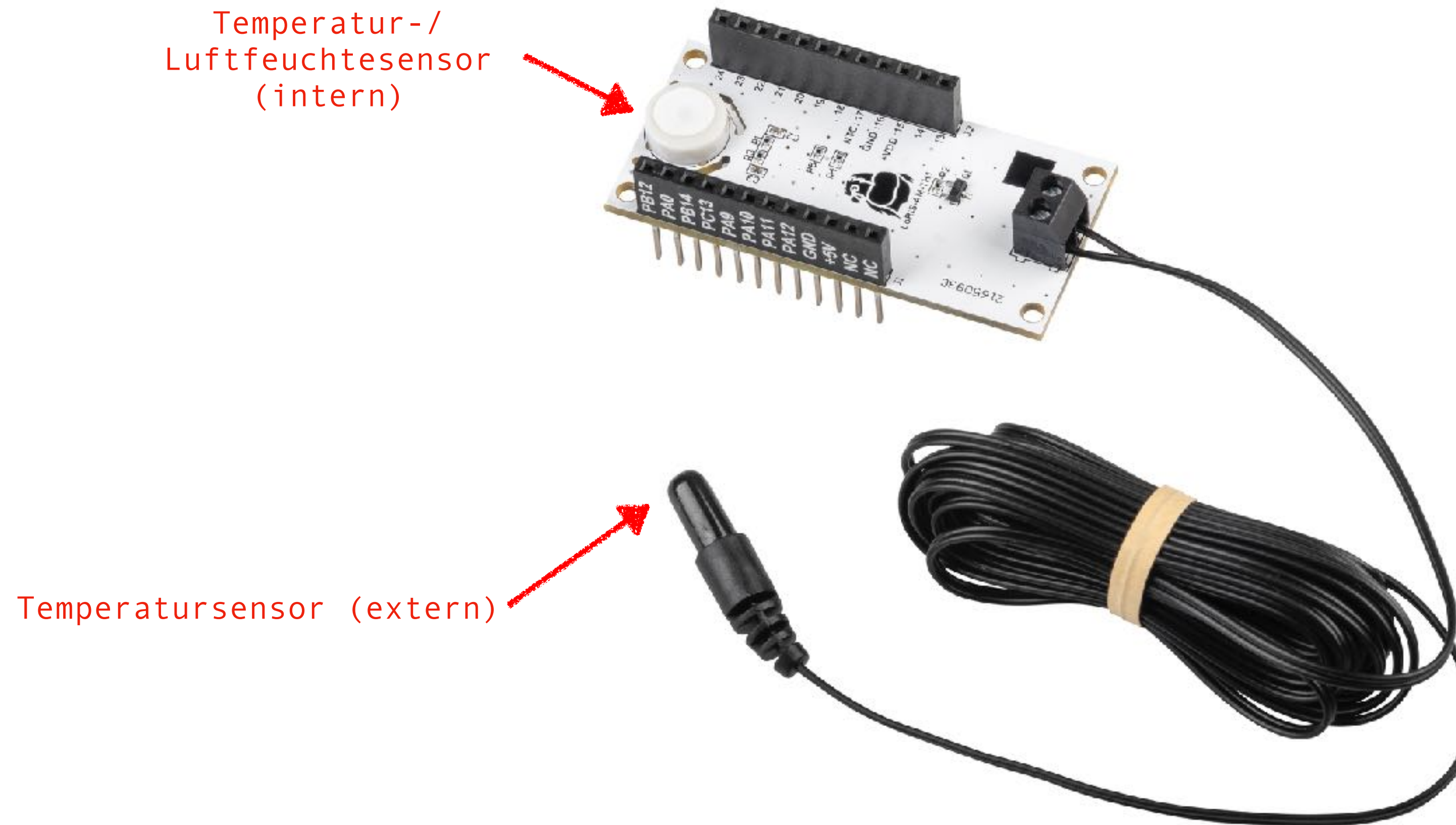
Downlink: LED steuern



LED an: 01 03 | LED aus: 01 02

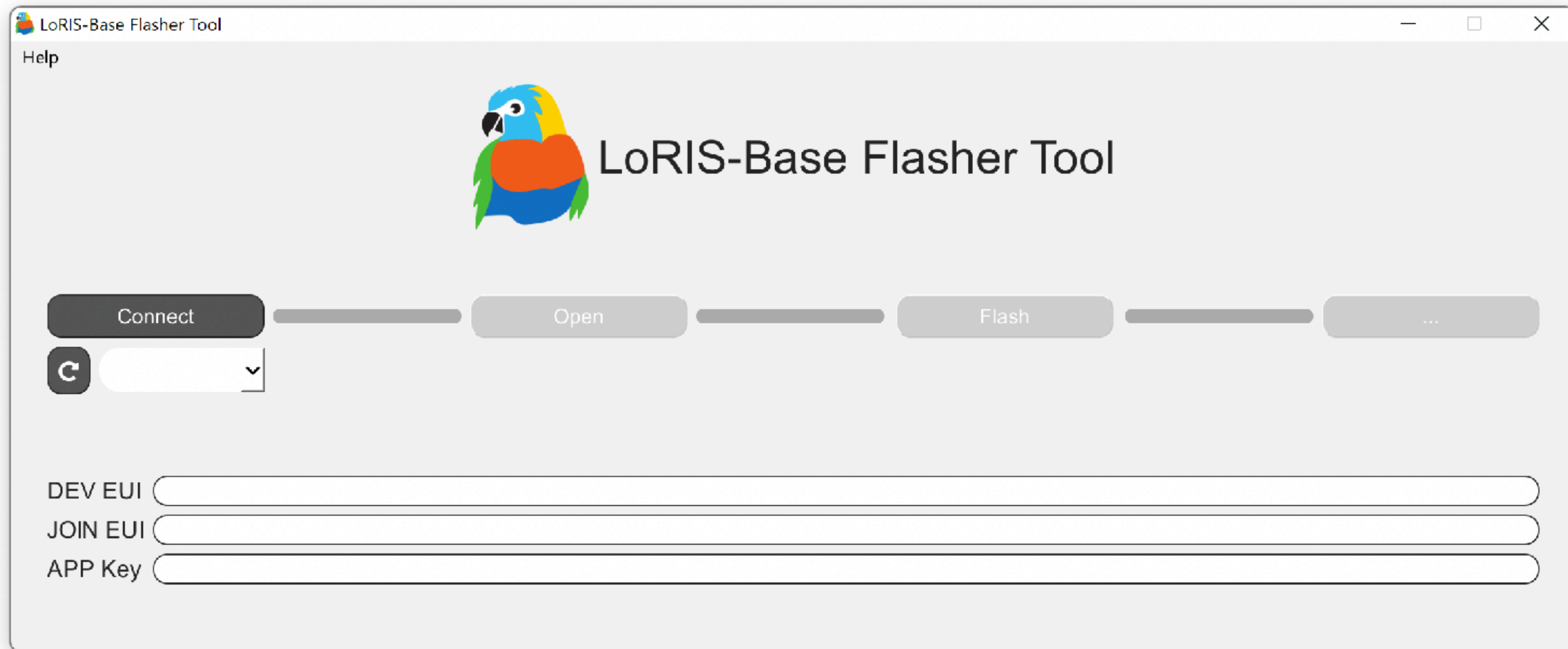
6. Temperatur und Luftfeuchte

Temperatur und Luftfeuchte Sensormodul



Temperatur und Luftfeuchte

Firmware aufspielen



Temperatur und Luftfeuchte

Messintervall einstellen

- Intervall voreingestellt auf 10 Minuten
- kann per Downlink-Nachricht geändert werden:

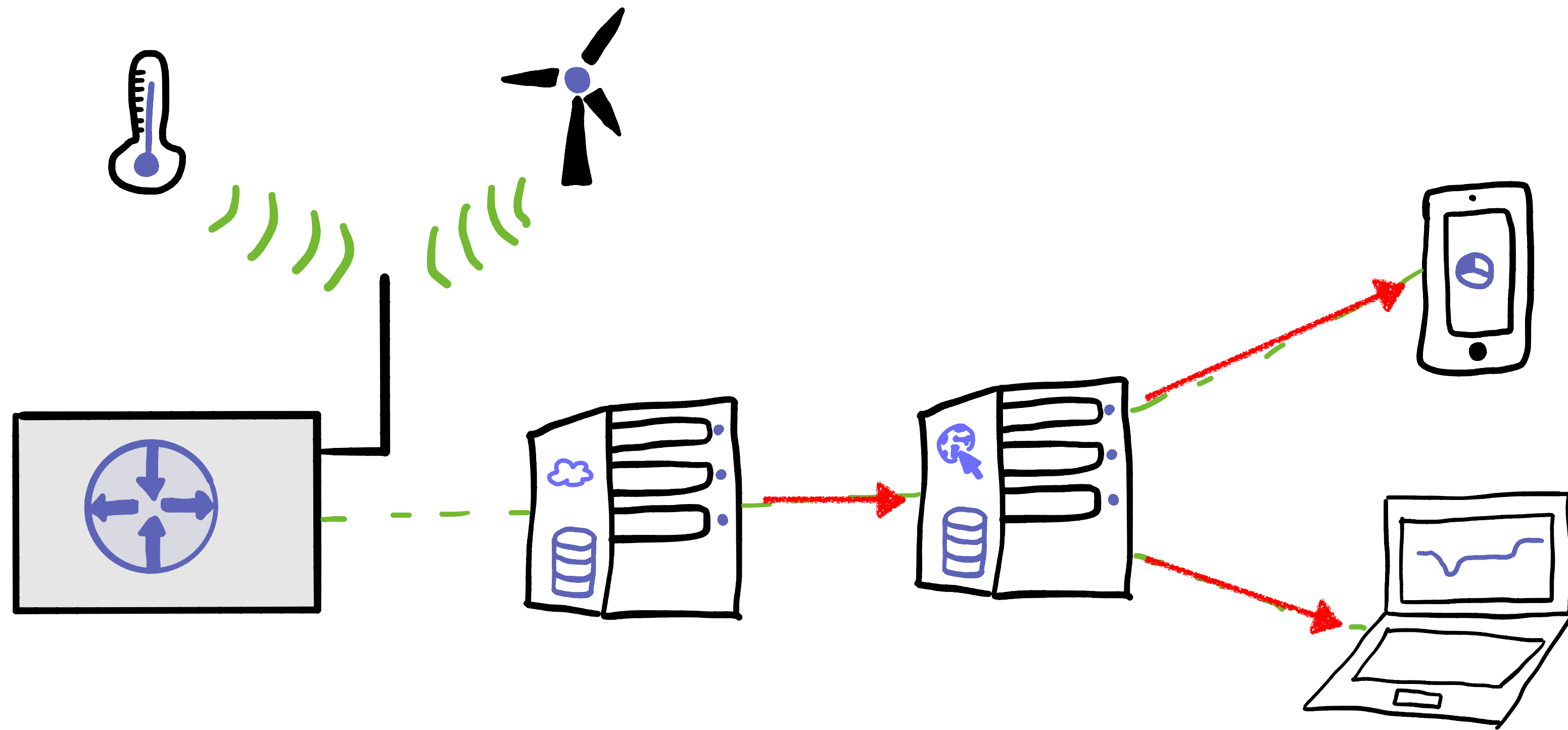
02 0A

Welches Board?
(hier: Temperatur/Luftfeuchte)

Sendeintervall (Hexadezimal)
(hier: 10 Minuten)

7. Datenweiterleitung

Datenweiterleitung

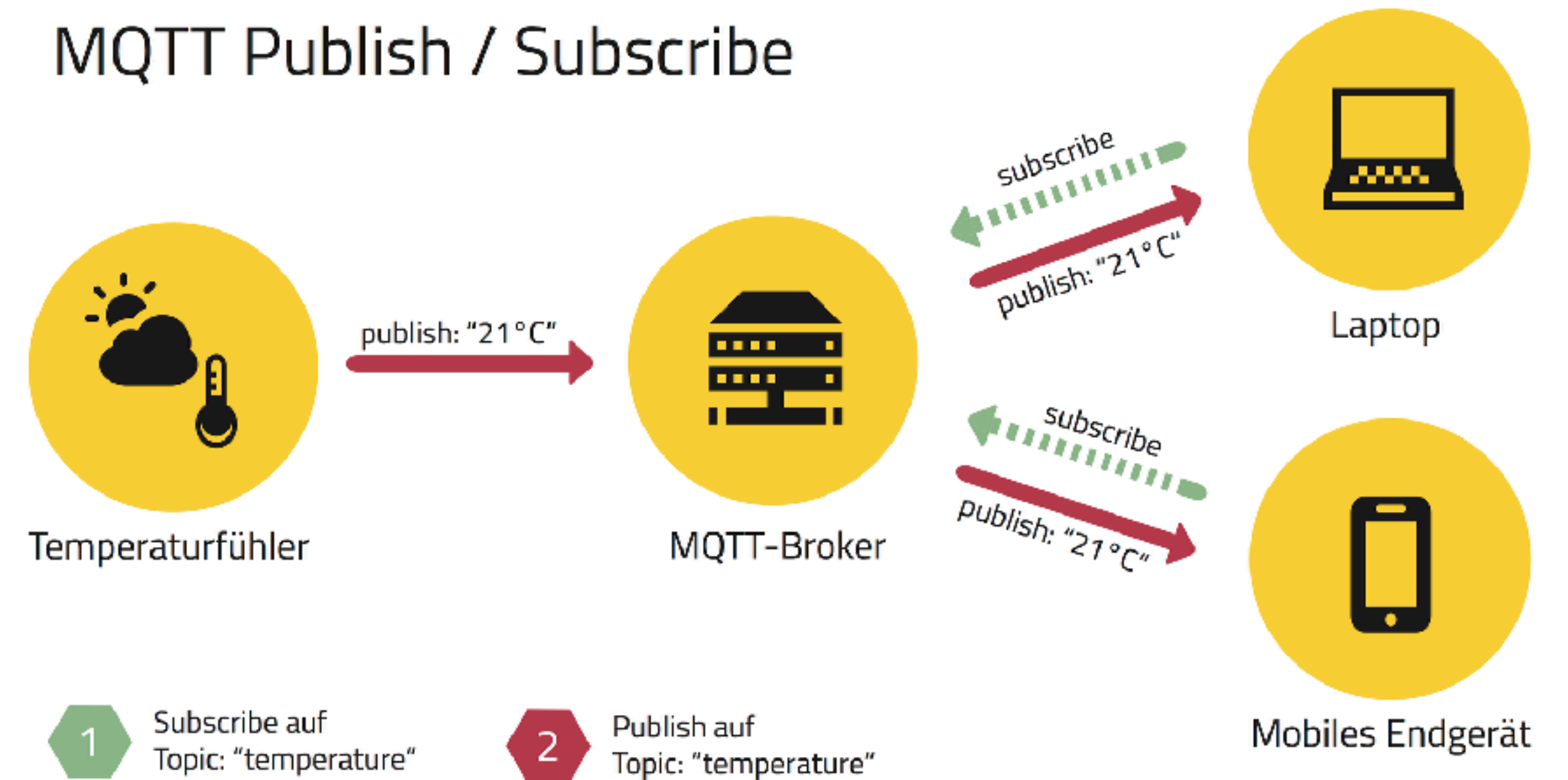


MQTT

Message Queue Telemetry Transport

- Maschine-Maschine Kommunikation
- 1999 zur Überwachung von Ölpipelines entwickelt
- Client-Server-Prinzip
- Publish/Subscribe - Architektur mit Topics
(z.B. Daheim/Wohnzimmer/Temperatur)

MQTT Publish / Subscribe



Datenweiterleitung

Datenweiterleitung über MQTT

The screenshot shows the MQTT configuration page in the The Things Stack Community Edition. The browser address bar displays `eu1.cloud.thethings.network`. The top navigation bar includes links for Overview, Applications (selected), Gateways, and Organizations. The left sidebar lists various components, with Integrations expanded and MQTT selected. The main content area is titled "MQTT" and provides an overview of the protocol, stating it is a publish/subscribe messaging protocol designed for IoT. It explains that every application on TTS automatically exposes an MQTT endpoint and that users need to create a new API key to connect. Further resources include links to the MQTT server and the official MQTT website. The "Connection information" section contains fields for the MQTT server host (Public address and Public TLS address), both set to `eu1.cloud.thethings.network:1883` and `eu1.cloud.thethings.network:8883` respectively. The "Connection credentials" section shows the Username as `loris-base-benno@ttn` and the Password field with a "Generate new API key" button and a "Go to API keys" link. The footer includes copyright information for 2022, language settings (EN), version (v3.18.1), and links to Documentation and Get support.

eu1.cloud.thethings.network

THE THINGS STACK Community Edition

Overview Applications Gateways Organizations

EU1 Community Fair use policy applies

bennohlz

Applications > LoRa Base > MQTT

MQTT

MQTT is a publish/subscribe messaging protocol designed for IoT. Every application on TTS automatically exposes an MQTT endpoint. In order to connect to the MQTT server you need to create a new API key, which will function as connection password. You can also use an existing API key, as long as it has the necessary rights granted.

Further resources

[MQTT server](#) | [Official MQTT website](#)

Connection information

MQTT server host

Public address `eu1.cloud.thethings.network:1883`

Public TLS address `eu1.cloud.thethings.network:8883`

Connection credentials

Username `loris-base-benno@ttn`

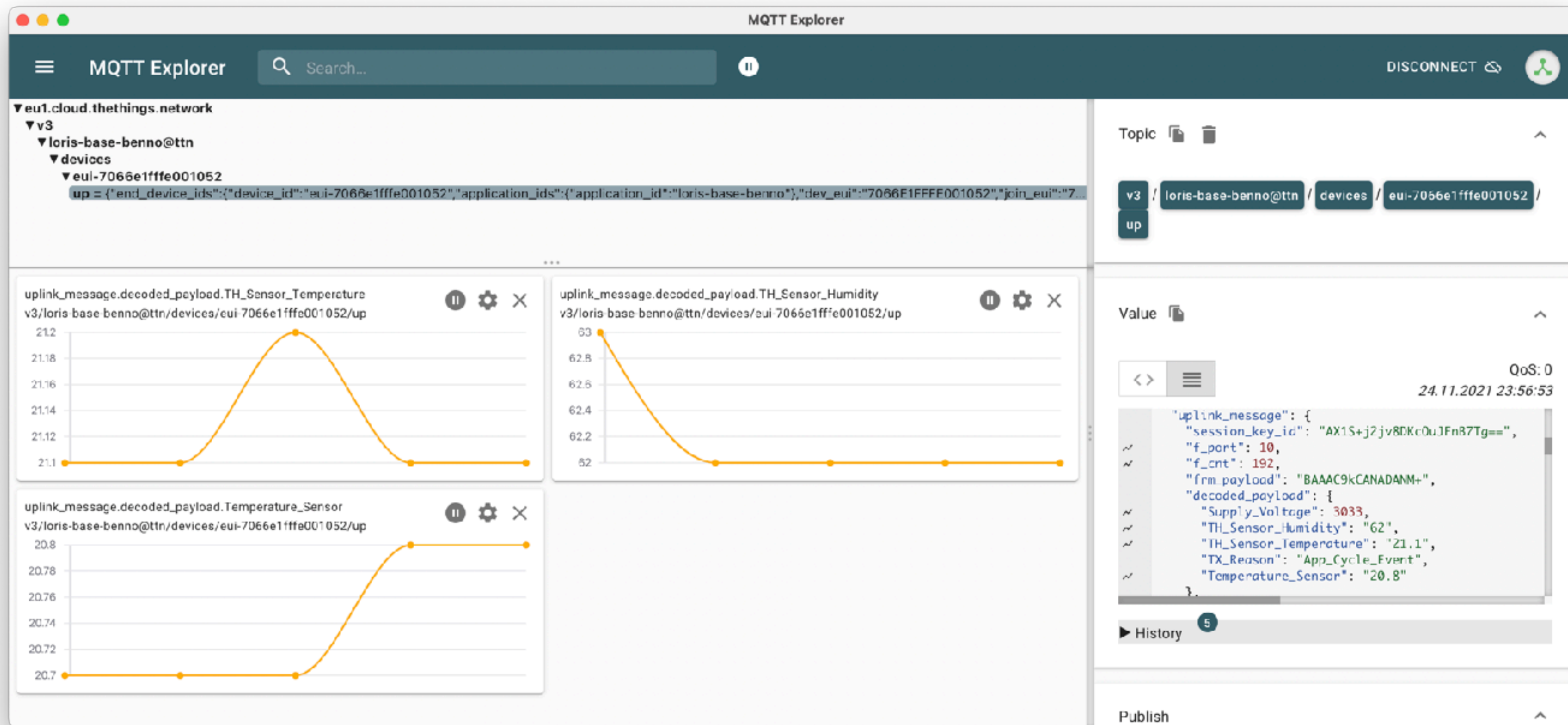
Password [Generate new API key](#) [Go to API keys](#)

© 2022 The Things Stack by The Things Network and The Things Industries

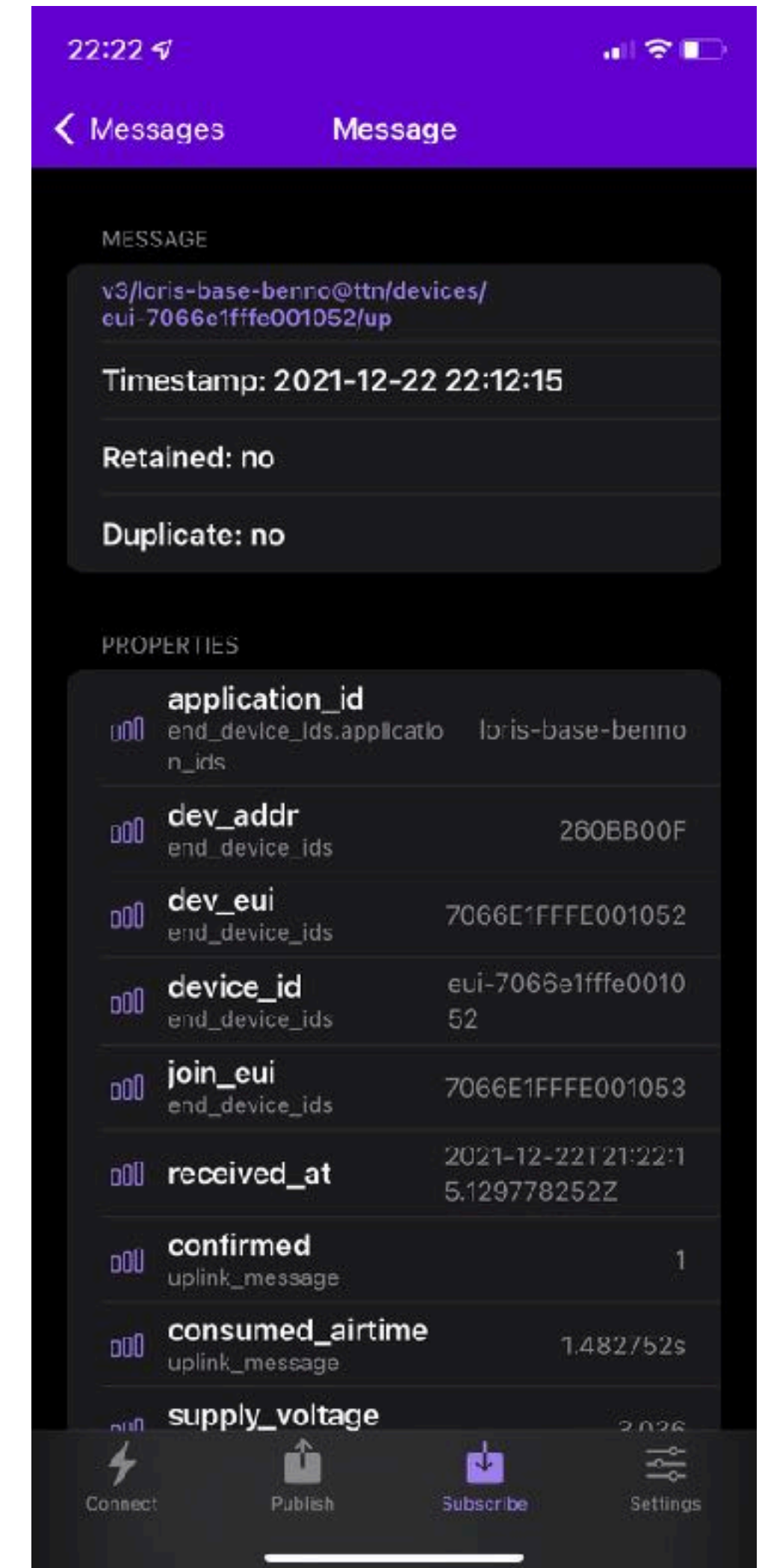
EN v3.18.1 Documentation [Get support](#)

Datenweiterleitung

MQTT-Clients



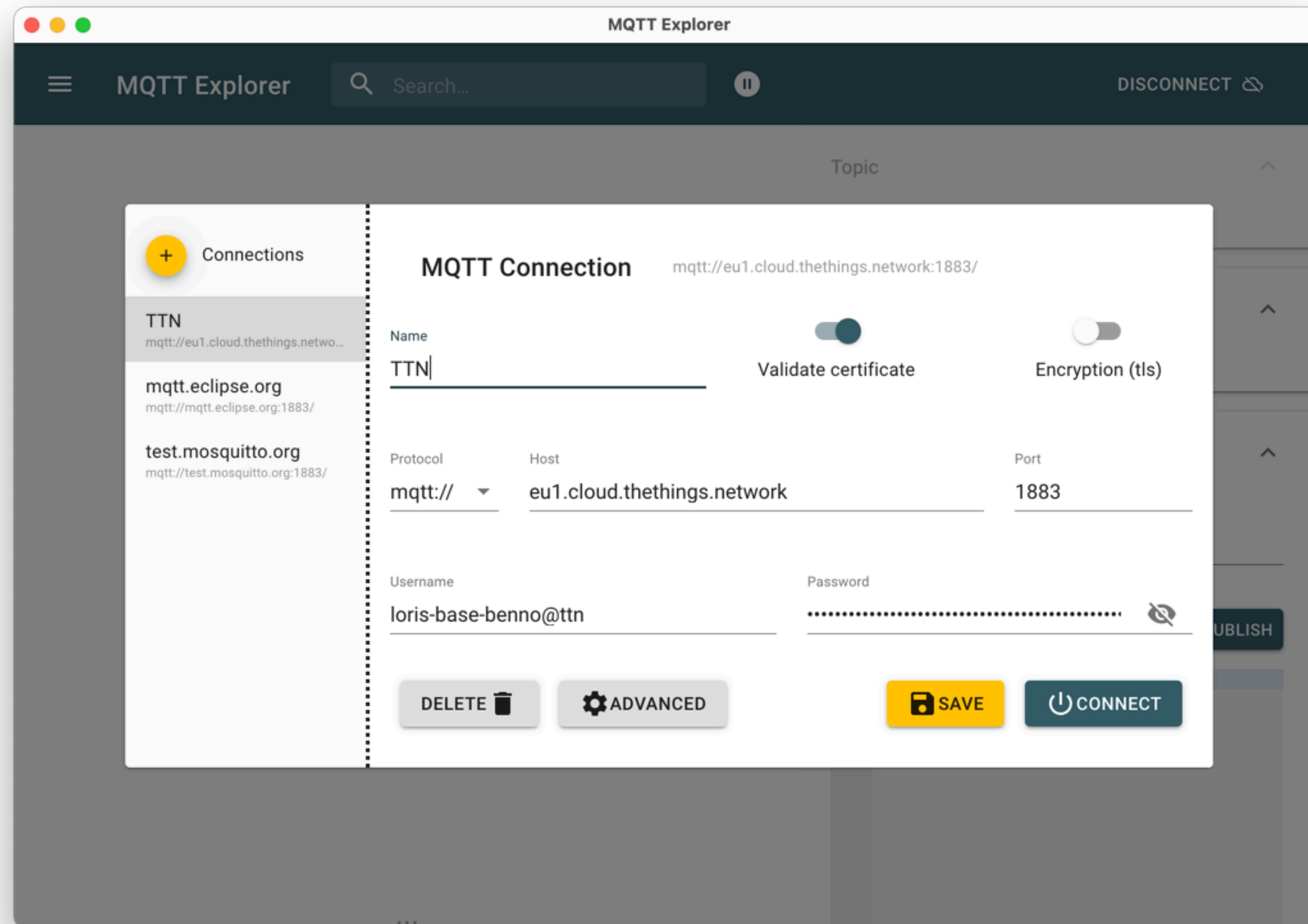
Die Desktop-Anwendung „MQTT Explorer“



Die iOS-App „EasyMQTT“

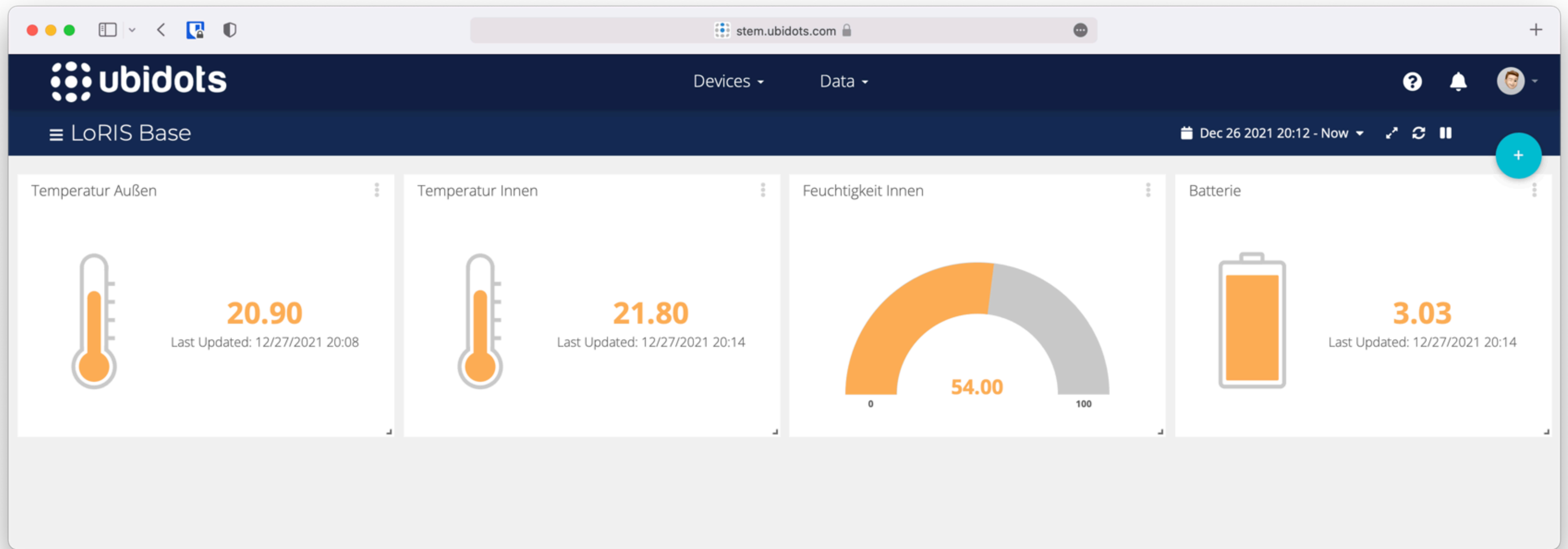
Datenweiterleitung

MQTT Explorer



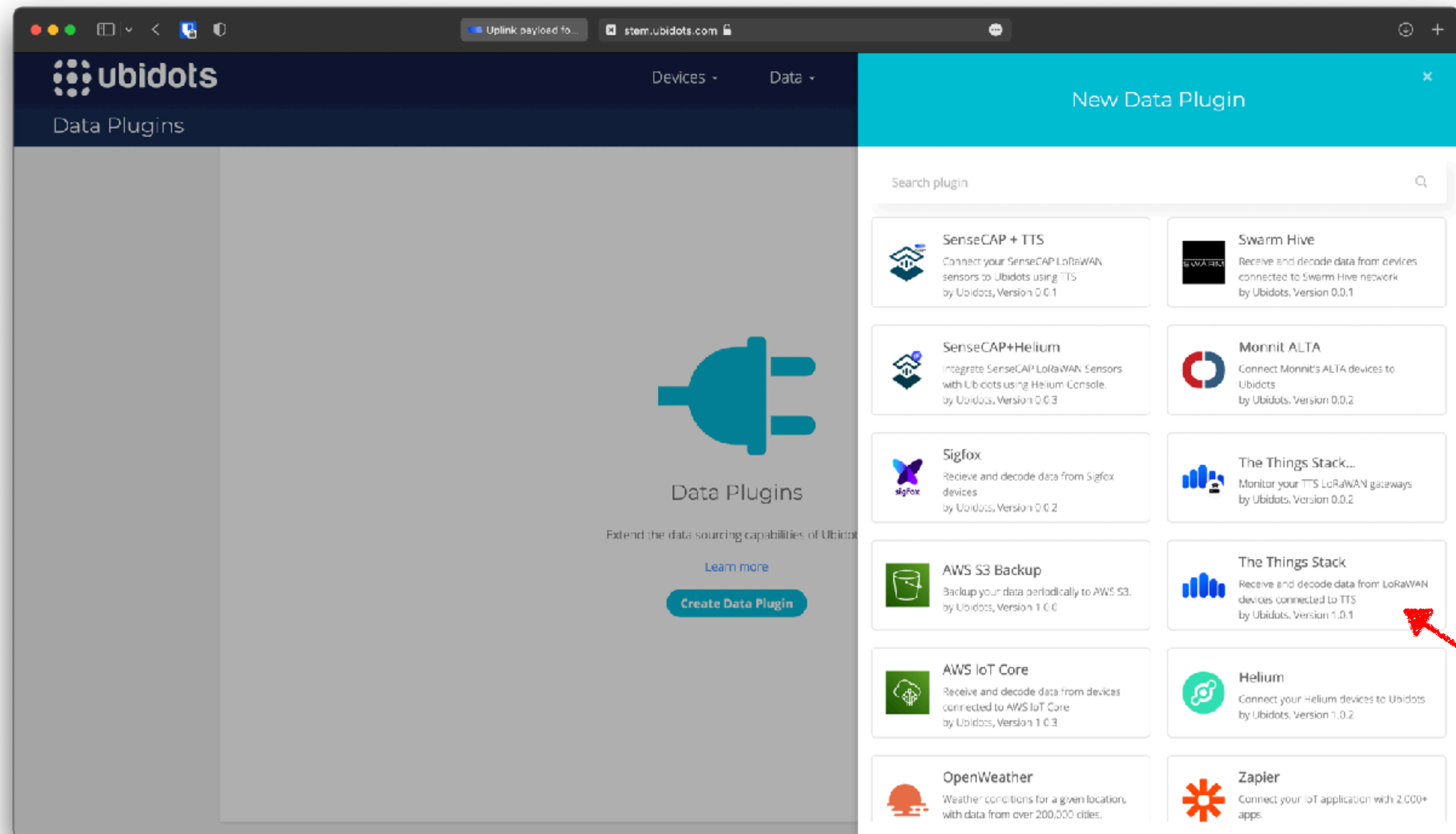
Datenweiterleitung

Datenweiterleitung und Visualisierung mit Ubidots



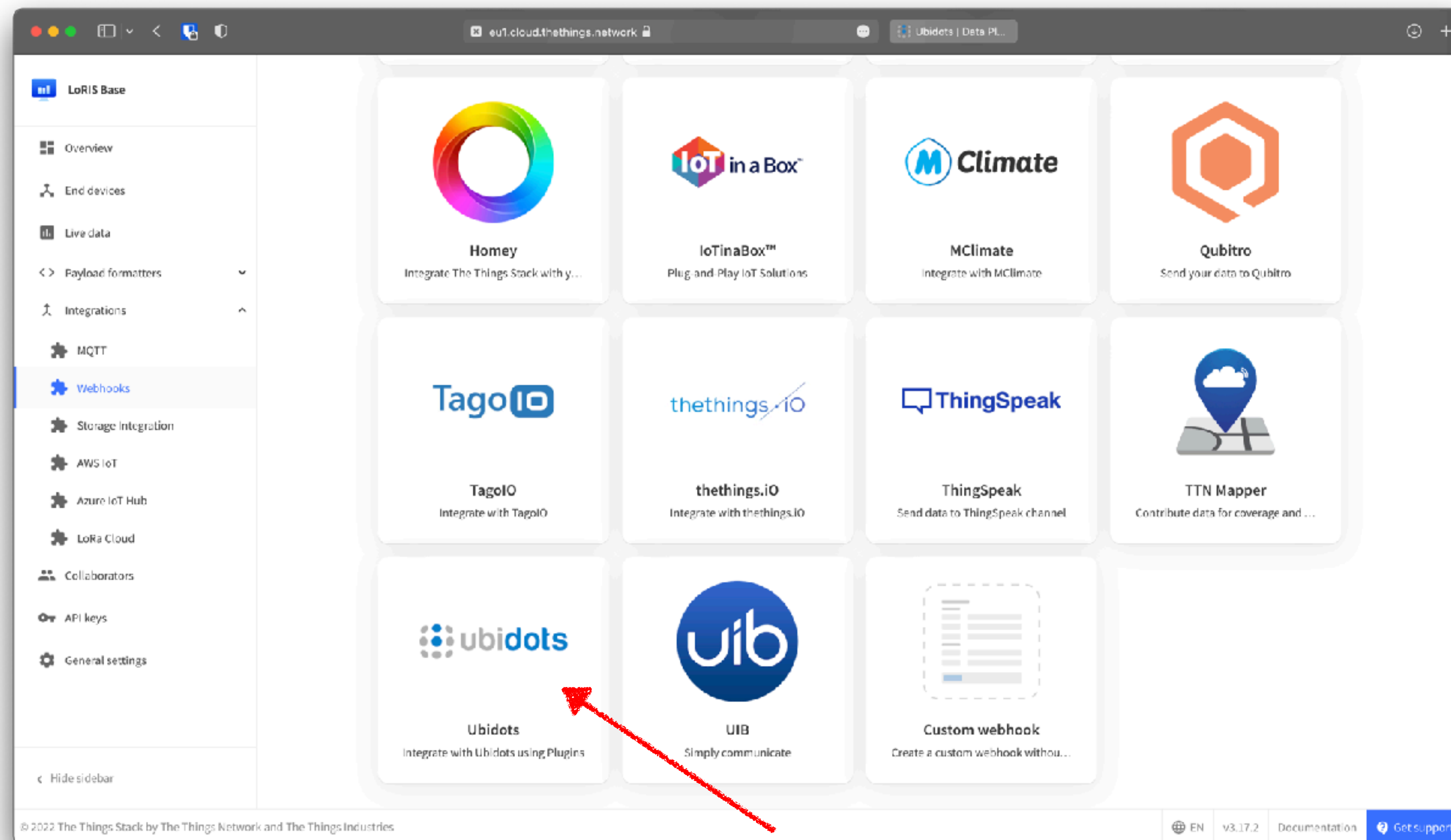
Datenweiterleitung

Datenweiterleitung und Visualisierung mit Ubidots



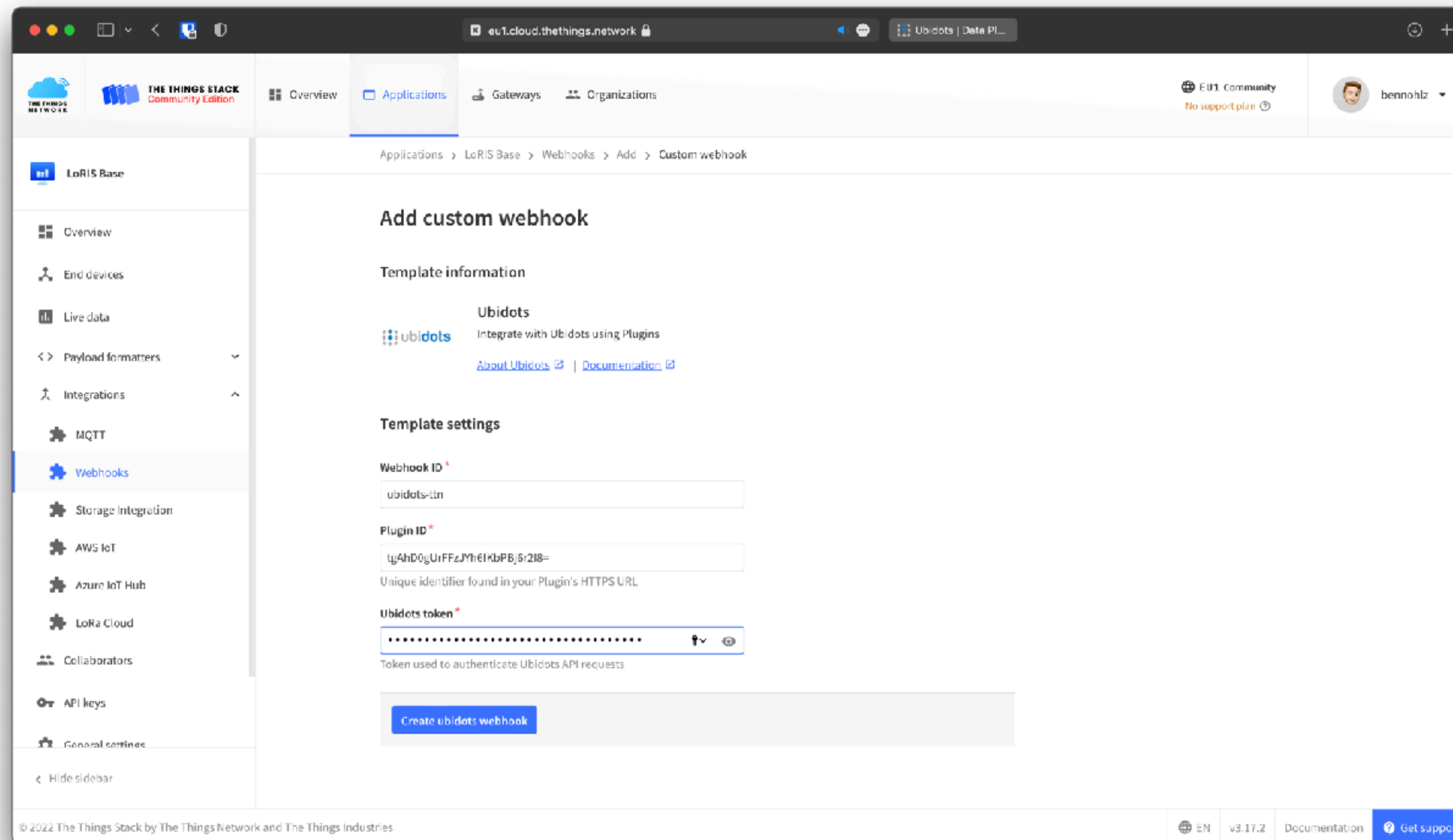
Datenweiterleitung

Datenweiterleitung und Visualisierung mit Ubidots



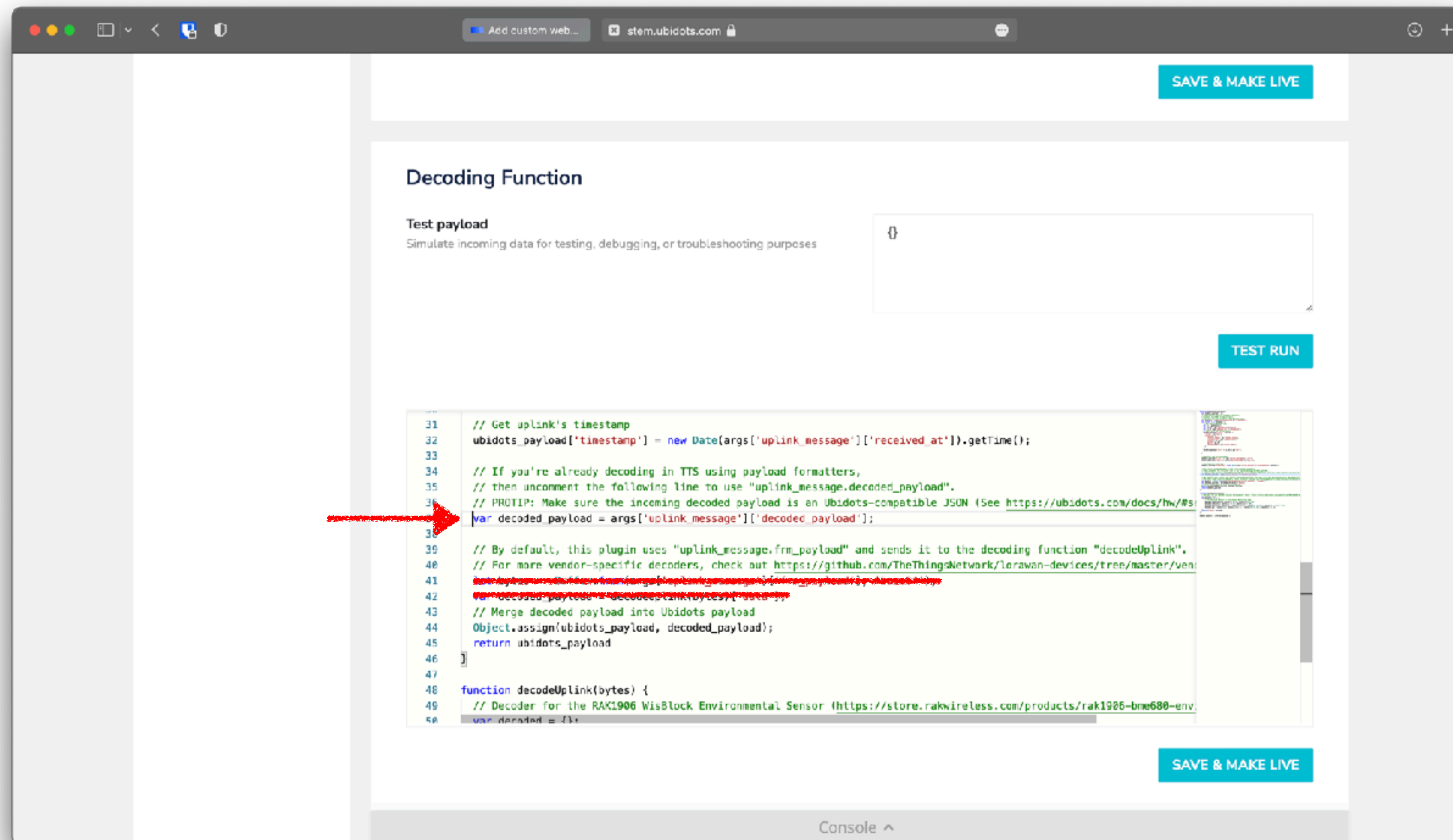
Datenweiterleitung

Datenweiterleitung und Visualisierung mit Ubidots



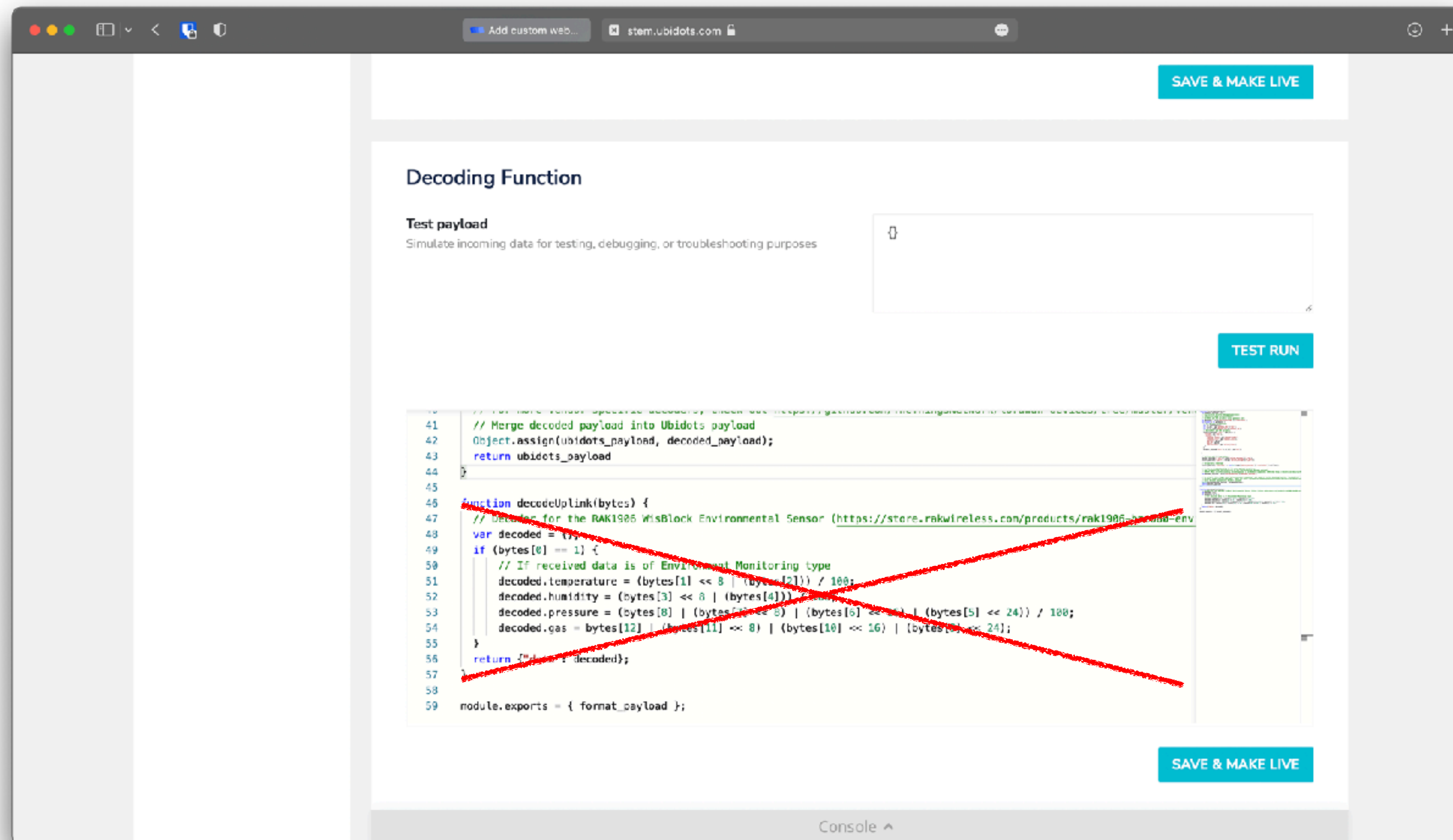
Datenweiterleitung

Datenweiterleitung und Visualisierung mit Ubidots



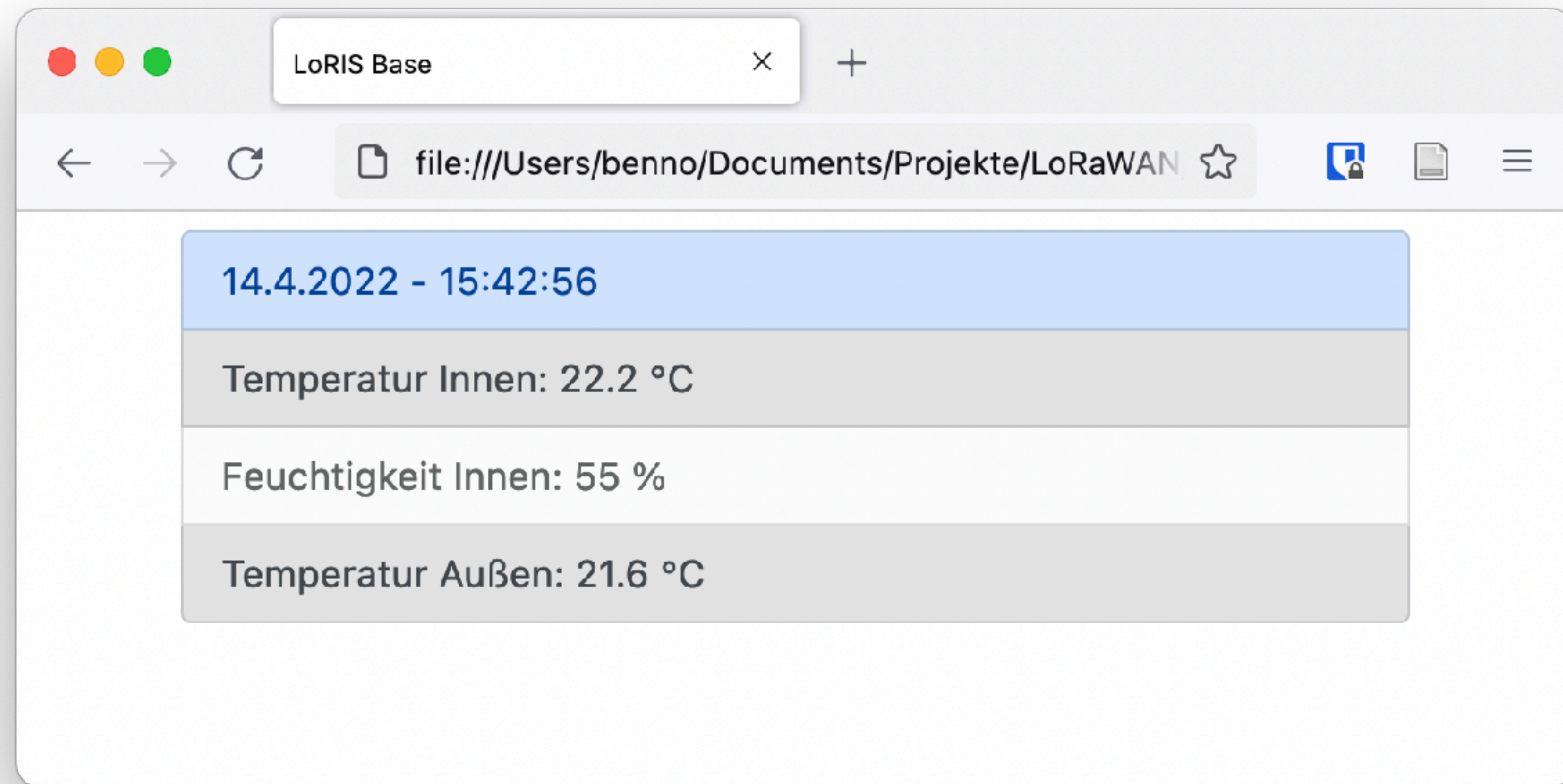
Datenweiterleitung

Datenweiterleitung und Visualisierung mit Ubidots



Datenweiterleitung

Datenabfrage via Javascript



Datenweiterleitung

Datenabfrage via Javascript

1. Daten-Speicherung mit **Storage-Integration**
2. Daten-Abruf über **Schnittstelle (API)**
3. Anzeige über **Javascript + Bootstrap** (Grafik-Framework)

Datenweiterleitung

Datenabfrage via Javascript

The screenshot displays the The Things Stack Community Edition web interface. The browser address bar shows `eu1.cloud.thethings.network`. The top navigation bar includes the The Things Network logo, the product name "THE THINGS STACK Community Edition", and tabs for Overview, Applications (selected), Gateways, and Organizations. On the right, it shows the "EU1 Community" logo with a note "Fair use policy applies" and a user profile for "bennohlz".

The left sidebar contains a menu for the "LoRIS Base" application, with options: Overview, End devices, Live data, Payload formatters, Integrations (selected), MQTT, Webhooks, Storage Integration (highlighted), AWS IoT, Azure IoT Hub, LoRa Cloud, Collaborators, API keys, and a "Hide sidebar" button.

The main content area is titled "Storage Integration" and shows the breadcrumb "Applications > LoRIS Base > Storage Integration". The text explains: "The Storage Integration allows storing received upstream messages in a persistent database, and retrieving them at a later time. This integration is implemented as an [Application Package](#) and can be enabled per application or per end device."

Under "Further resources", there are links for [Storage Integration](#) and [Application Packages](#).

The "Status" section indicates: "The Storage Integration is currently activated". It provides instructions to use the endpoints below to retrieve data from the storage, with a link to the [Storage Integration API](#).

Two API endpoints are listed:

```
GET https://eu1.cloud.thethings.network/api/v3/as/applications/loris-base-benno/packages/storage/{type}
```

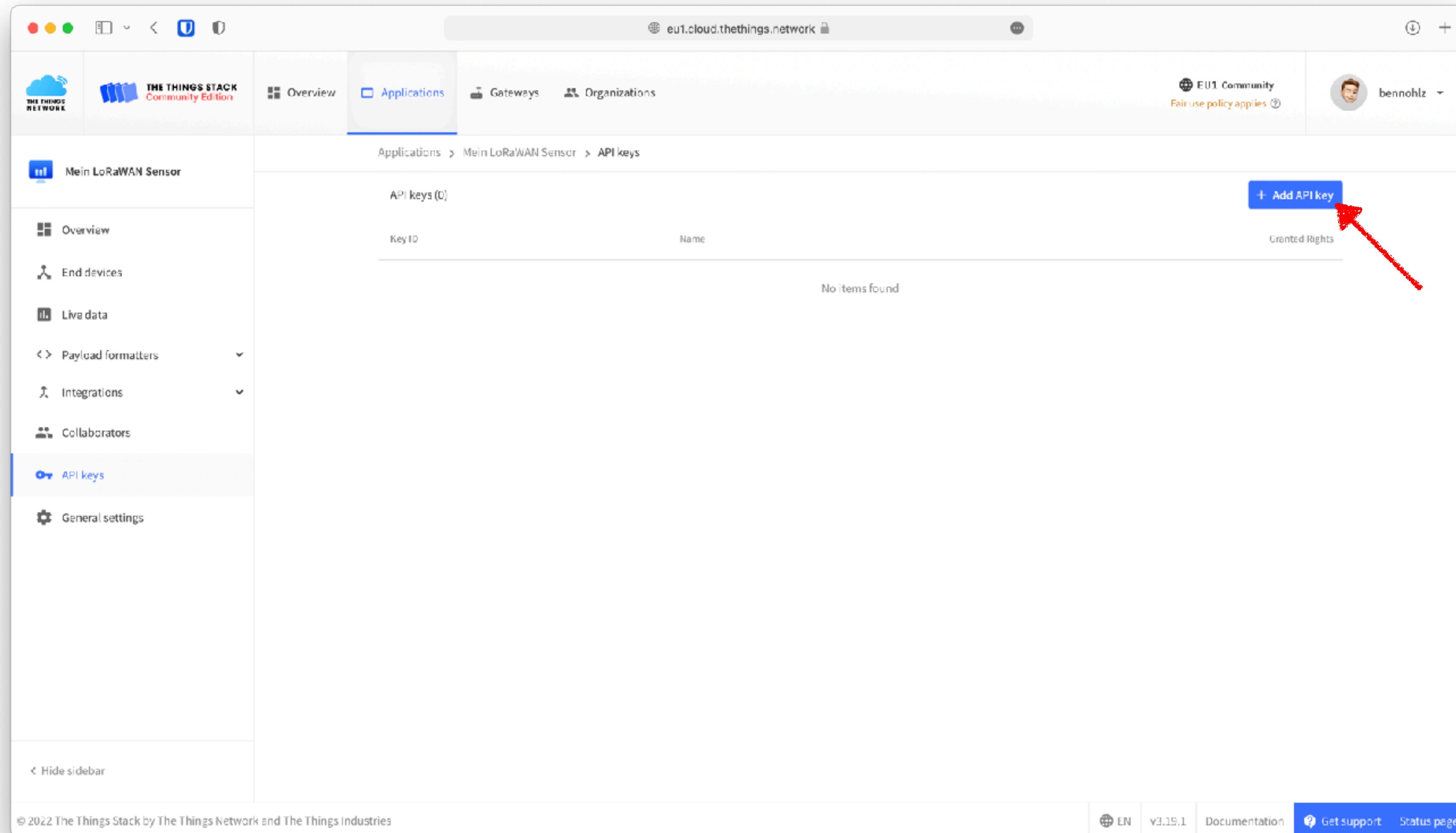
```
GET https://eu1.cloud.thethings.network/api/v3/as/applications/loris-base-benno/devices/{device_id}/packages/storage/{type}
```

A red button labeled "Deactivate Storage Integration" is visible.

The footer contains copyright information: "© 2022 The Things Stack by The Things Network and The Things Industries", along with links for language (EN), version (v3.15.2), documentation, and support.

Datenweiterleitung

Datenabfrage via Javascript



Datenweiterleitung

Datenabfrage via Javascript

The screenshot shows the 'Integrations' section of the The Things Stack v3.18.2 documentation. The left sidebar lists various integration options, with 'Retrieve Messages' highlighted under 'Storage Integration'. The main content area is titled 'Retrieve Uplinks using the HTTP API' and includes a note about required API rights. It provides two curl commands for retrieving uplinks for an application or a specific device. Below the commands, an example API response is shown, and a note explains how to use field masks. A red arrow points to the second curl command. The bottom right corner features a chat bubble icon.

Storage Integrati... thethingsindustries.com

THE THINGS STACK v3.18.2 Getting Started Devices Gateways Integrations Reference Search the docs... Get The Things Stack

Integrations

- Overview
- Adding Applications
- Adding Integrations
- Cloud Integrations
- MQTT Server
- Pub/Sub
- LoRa Cloud
- Node-RED
- IFTTT
- Payload Formatters
- Storage Integration
 - Configuration
 - Enable/Disable
 - Retrieve Messages
- Webhooks

Retrieve Uplinks using the HTTP API

A valid API key with the `RIGHT_APPLICATION_TRAFFIC_READ` rights is required.

```
# Retrieve stored uplinks for application "appl"
curl -G "https://thethings.example.com/api/v3/as/applications/<application-id>/packages/storage/uplink_m" \
  -H "Authorization: Bearer $API_KEY" \
  -H "Accept: text/event-stream" \
  -d "limit=10" \
  -d "after=2020-08-20T00:00:00Z"

# Retrieve stored uplinks for device "dev1" of application "appl"
curl -G "https://thethings.example.com/api/v3/as/applications/<application-id>/devices/<device-id>/packa" \
  -H "Authorization: Bearer $API_KEY" \
  -H "Accept: text/event-stream" \
  -d "limit=10" \
  -d "after=2020-08-20T00:00:00Z"
```

▼ See example API response

```
{ "result": { "end_device_ids": { "device_id": "dev1", "application_ids": { "application_id": "appl", "dev_eui": "..." } } } }
{ "result": { "end_device_ids": { "device_id": "dev1", "application_ids": { "application_id": "appl", "dev_eui": "..." } } } }
```

Use field masks to specify a subset of fields that should be returned by the API. For example, to retrieve the decoded payload field only, set the field mask as shown below. Note that the end device identifiers and the timestamp are always included in the message.

```
curl -G "https://thethings.example.com/api/v3/as/applications/<application-id>/packages/storage/uplink_m" \
  -H "Authorization: Bearer $API_KEY" \
  -H "Accept: text/event-stream" \
  -d "limit=10" \
  -d "after=2020-08-20T00:00:00Z" \
  -d "field_mask=up.uplink_message.decoded_payload"
```

On this page

- Retrieve Uplinks using the HTTP API
- Retrieve Uplinks using the CLI

Datenweiterleitung

Datenabfrage via Javascript

- HTML mit Bootstrap als Grafik-Framework:
<https://getbootstrap.com/docs/5.1/getting-started/introduction/>
- Javascript-Code für Datenabruf und -anzeige



index.html

Heute, 11:23

1 KB

HTML-Dokument



mainScript.js

14.04.22, 15:57

2 KB

JavaScript script

Datenweiterleitung

Datenabfrage via Javascript

Antwort (im JSON-Format):

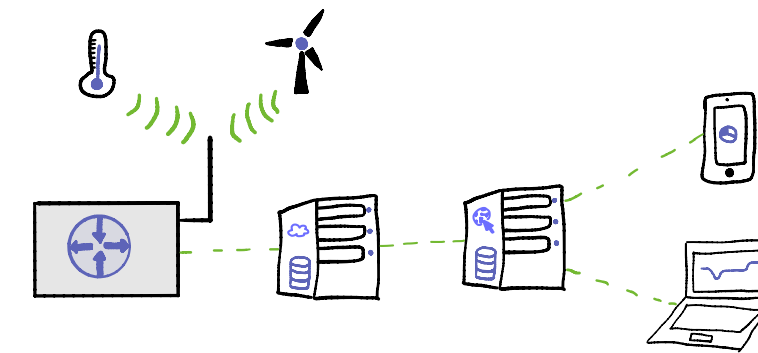
```
▼ result:
  ▼ end_device_ids:
    device_id: "eui-7066e1fffe001052"
    ▼ application_ids:
      application_id: "loris-base-benno"
    dev_eui: "7066E1FFFE001052"
    dev_addr: "260B2B6A"
    received_at: "2022-04-14T13:53:08.502554369Z"
    ▼ uplink_message:
      f_port: 10
      f_cnt: 5892
      frm_payload: "BAAAC1wCANKDA0E4"
      ▼ decoded_payload:
        Supply_Voltage: 2908
        TH_Sensor_Humidity: "56"
        TH_Sensor_Temperature: "22.5"
        TX_Reason: "App_Cycle_Event"
        Temperature_Sensor: "21.7"

▼ rx_metadata:
  ▼ 0:
    ▼ gateway_ids:
      gateway_id: "sfz-ox-gateway"
      eui: "DCA632FFFE0CBFCA"
      time: "2022-04-14T13:53:08.255Z"
      timestamp: 3513463043
      rssi: -119
      channel_rssi: -119
      snr: 3.2
    ▼ location:
      latitude: 48.06489508067312
      longitude: 9.954755902290346
      altitude: 541
      source: "SOURCE_REGISTRY"
  ▼ 1:
    ▼ gateway_ids:
      gateway_id: "testttn"
      eui: "3133303747003B00"
      time: "2022-04-14T13:53:07.965870Z"
      timestamp: 1264820596
      rssi: -110
      channel_rssi: -110
      snr: -1.75

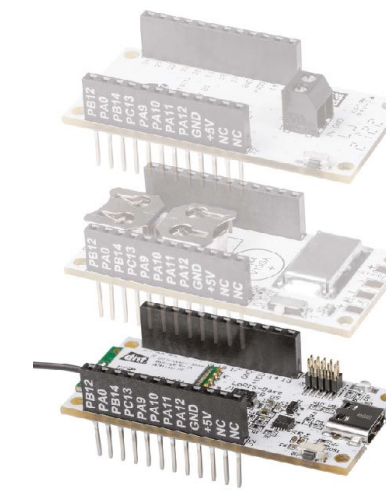
▼ settings:
  ▼ data_rate:
    ▼ lora:
      bandwidth: 125000
      spreading_factor: 7
    coding_rate: "4/5"
    frequency: "868100000"
    timestamp: 3513463043
    time: "2022-04-14T13:53:08.255Z"
    received_at: "2022-04-14T13:53:08.292999194Z"
    consumed_airtime: "0.061696s"
  ▼ network_ids:
    net_id: "000013"
    tenant_id: "ttn"
    cluster_id: "ttn-eu1"
```

Was haben wir gelernt?

- Was ist **LoRaWAN** und wie funktioniert es?



- Wie kann ich einen **Sensor** ins Netzwerk **einbinden**?



- Wie können die Daten **gespeichert und angezeigt** werden?

