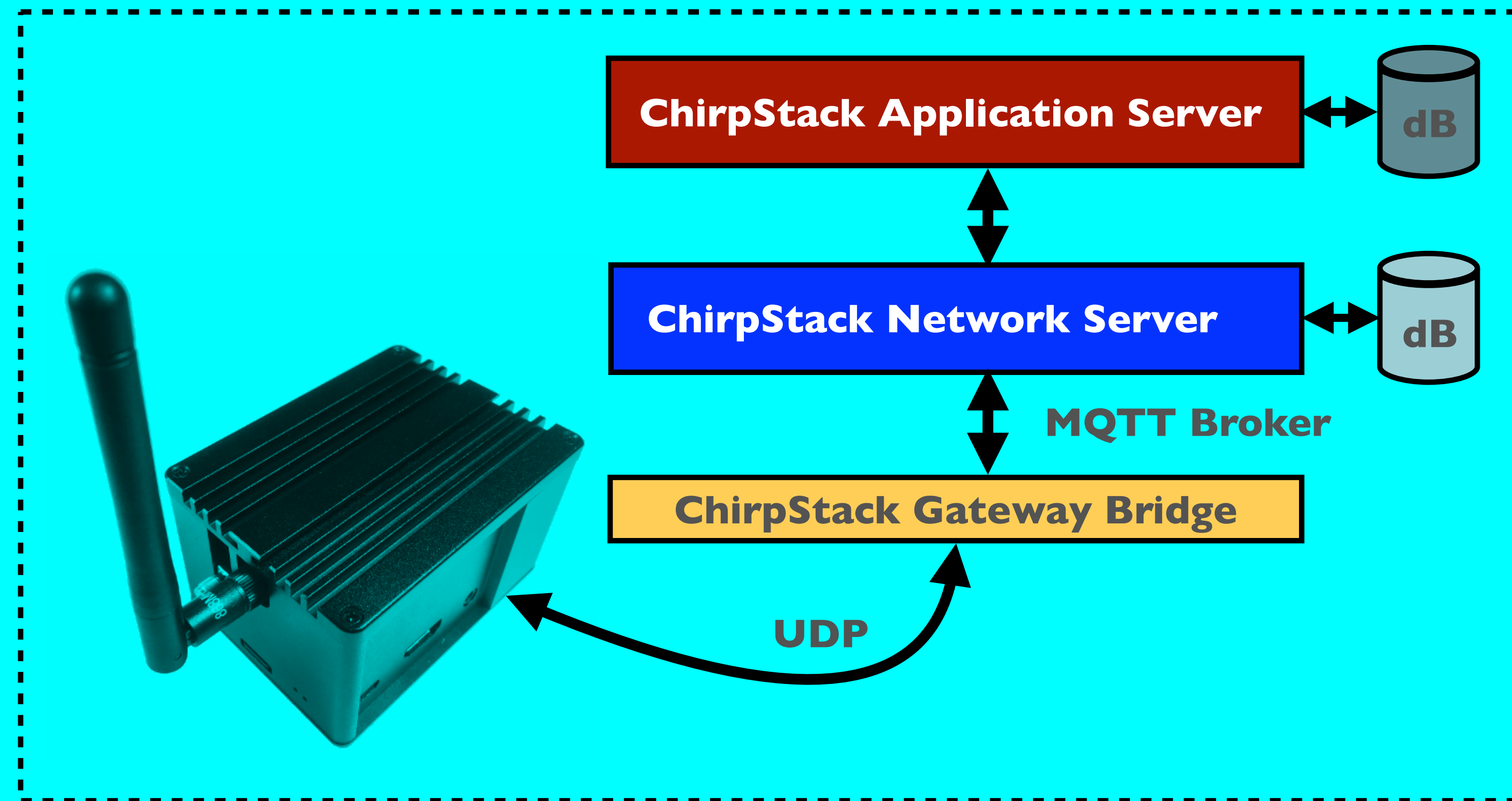


LORA / LORAWAN TUTORIAL 3 I

ChirpStack



INTRO

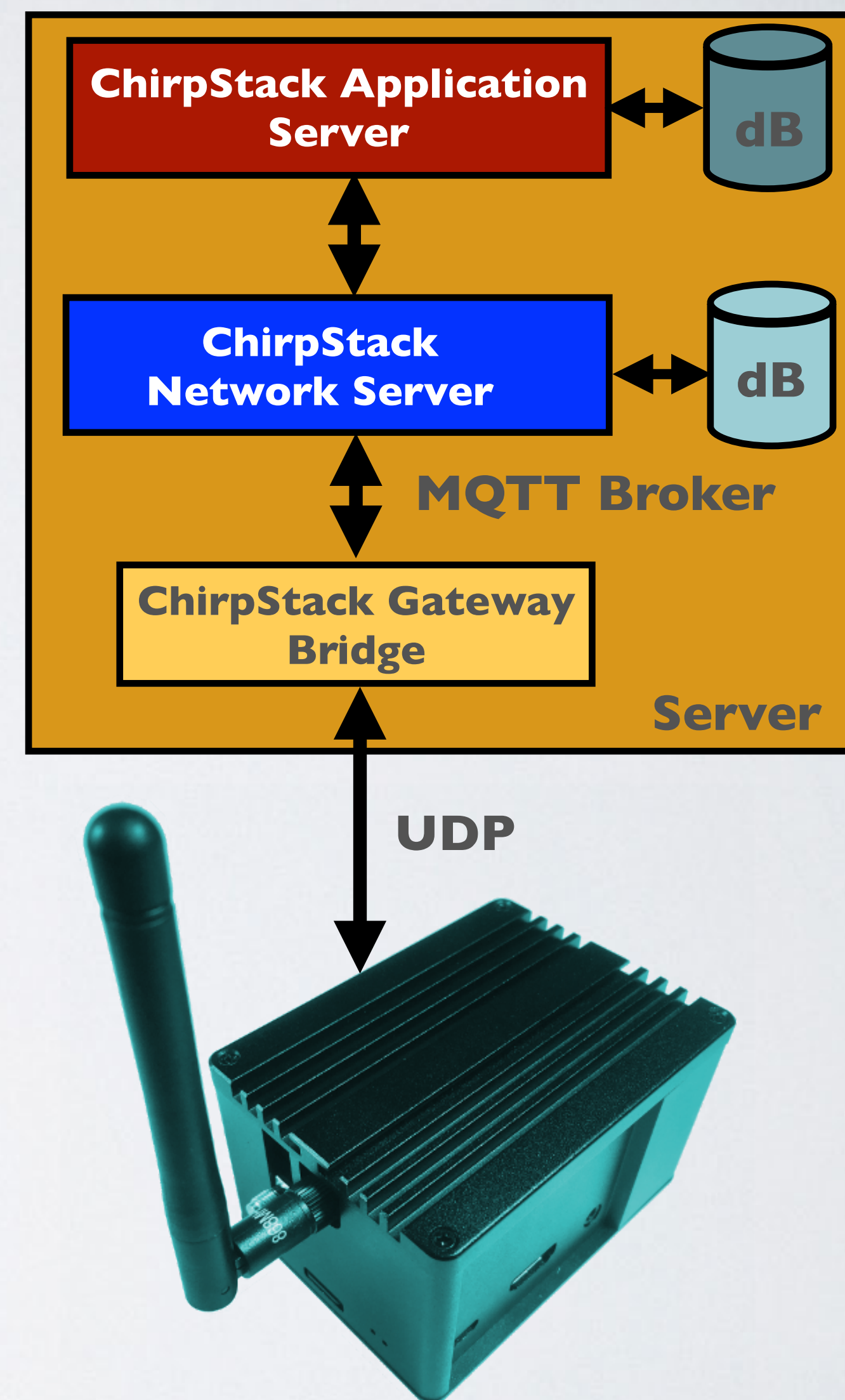
- In this tutorial I will demonstrate how to install and configure the open source LoRaWAN network server called ChirpStack.

WHAT IS CHIRPSTACK

- ChirpStack provides open source components for building LoRaWAN networks.
- In Nov 2019 the LoRa Server was renamed to ChirpStack due to trademark conflicts.
- The author of ChirpStack is Orne Brocaar, he started the project end 2015 as a side project and the project has grown significantly thanks to sponsoring.
- ChirpStack consists of four major components: ChirpStack Gateway Bridge, ChirpStack Network Server, ChirpStack Application Server and ChirpStack Geolocation Server. The Geolocation Server will not be discussed in this tutorial.
- All components are licensed under the MIT license and can be used for commercial purposes.

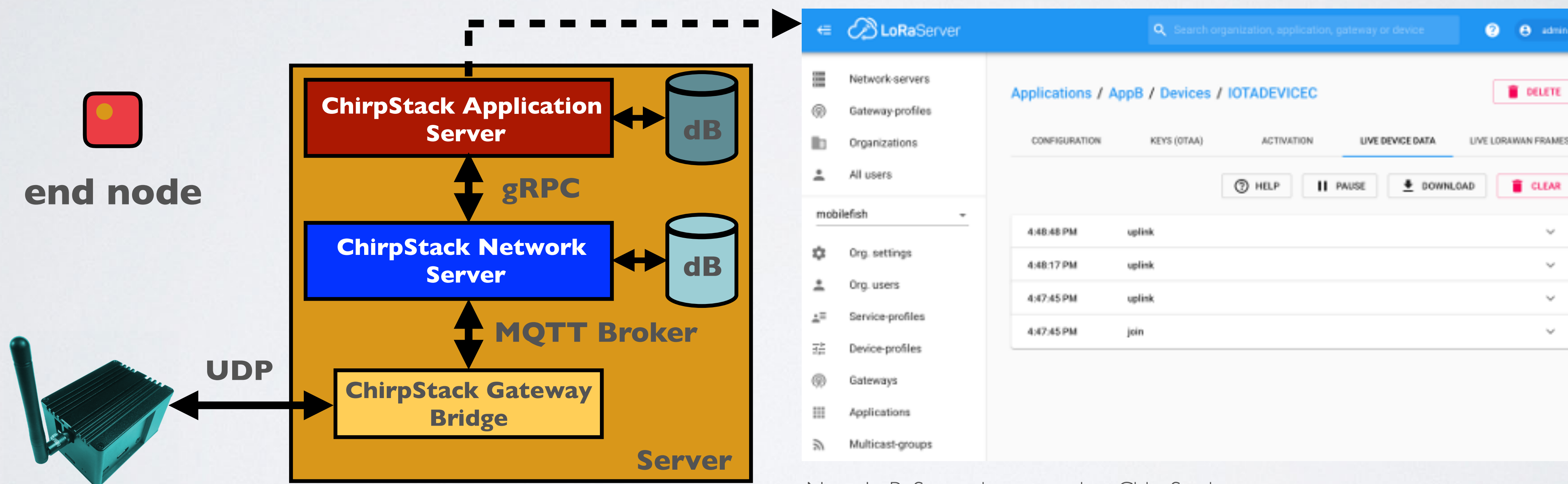
CHIRPSTACK

- The **ChirpStack Gateway Bridge** abstracts the packet forwarder UDP protocol data into JSON and sends it to the ChirpStack Network Server over MQTT.
- The **ChirpStack Network Server** is the LoRaWAN network server. It de-duplicates and handles the received uplink frames from the gateway(s), handles the LoRaWAN mac layer and schedules the downlink data transmissions.
- The **ChirpStack Application Server** is the LoRaWAN Application Server and handles the join-requests, encryption of application payloads and offers a RESTful JSON API, gRPC API or MQTT for external services.



CHIRPSTACK

- The **ChirpStack Application Server** has a web interface to manage users, organisations, applications and devices. It also allows you to see the received sensor data.



Note: LoRaServer is renamed to ChirpStack

CHIRPSTACK

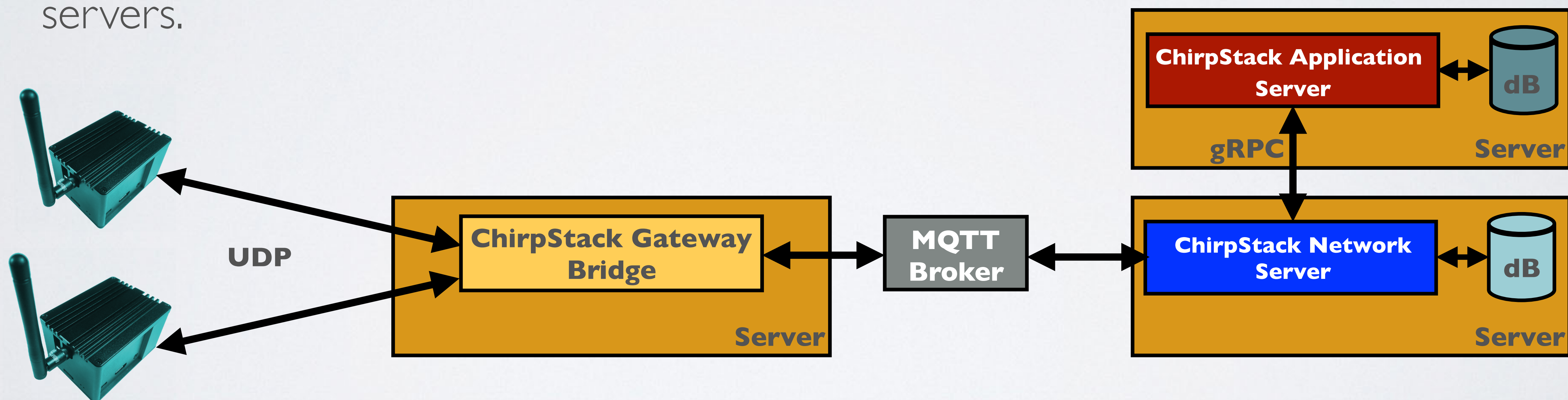
- Many users are sending their sensor data to The Things Network.
- Now you can create your own private LoRaWAN network by using the ChirpStack components.

CHIRPSTACK INFORMATION

- More information about the ChirpStack:
<https://www.chirpstack.io>
- All ChirpStack project code:
<https://github.com/brocaar>
- ChirpStack forum:
<https://forum.chirpstack.io>

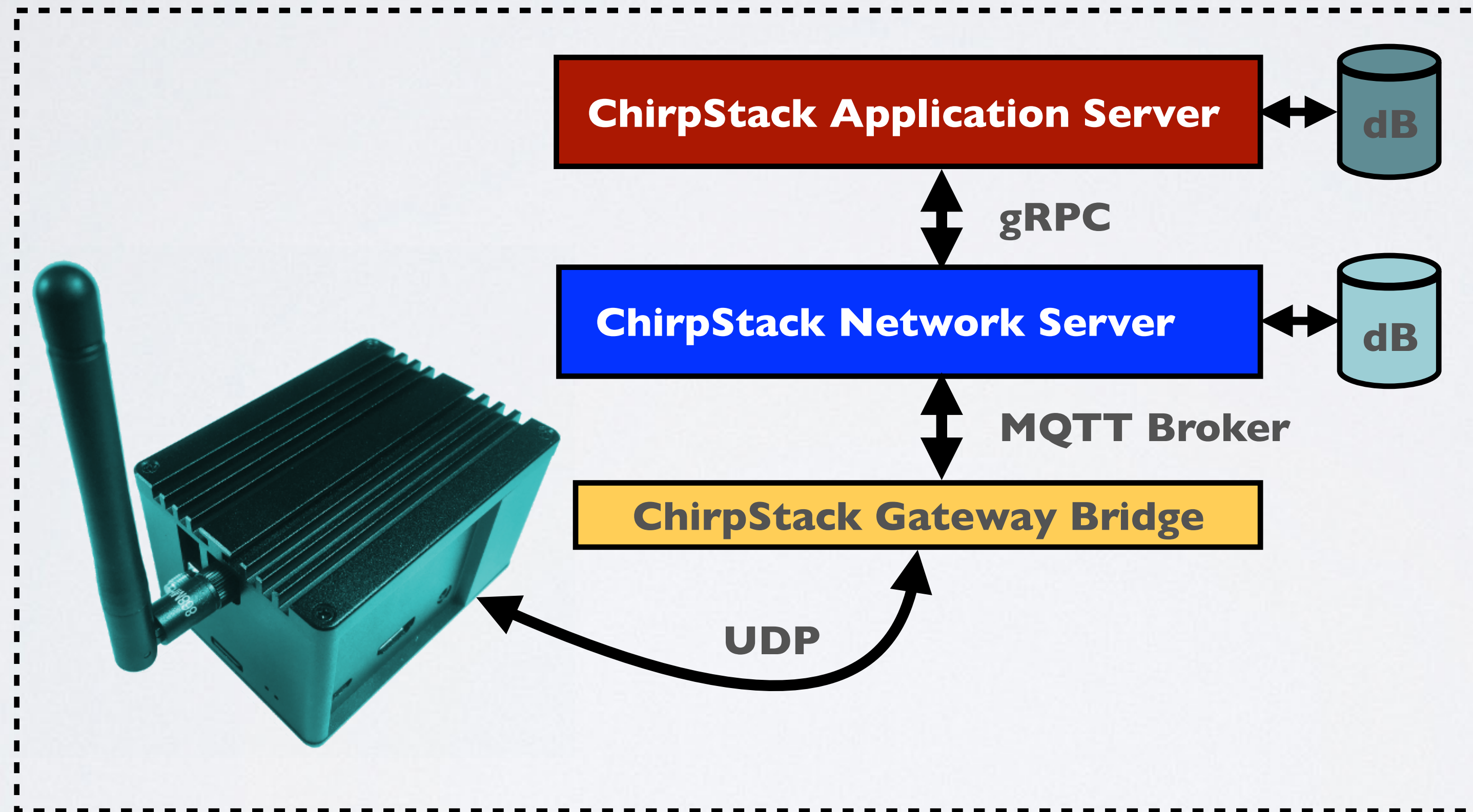
CHIRPSTACK COMPONENTS

- The ChirpStack Network Server and the ChirpStack Application Server each requires their own PostgreSQL database.
- There are many configurations of these components possible, for example:
 - All ChirpStack components and their dependencies can be installed on their own servers.



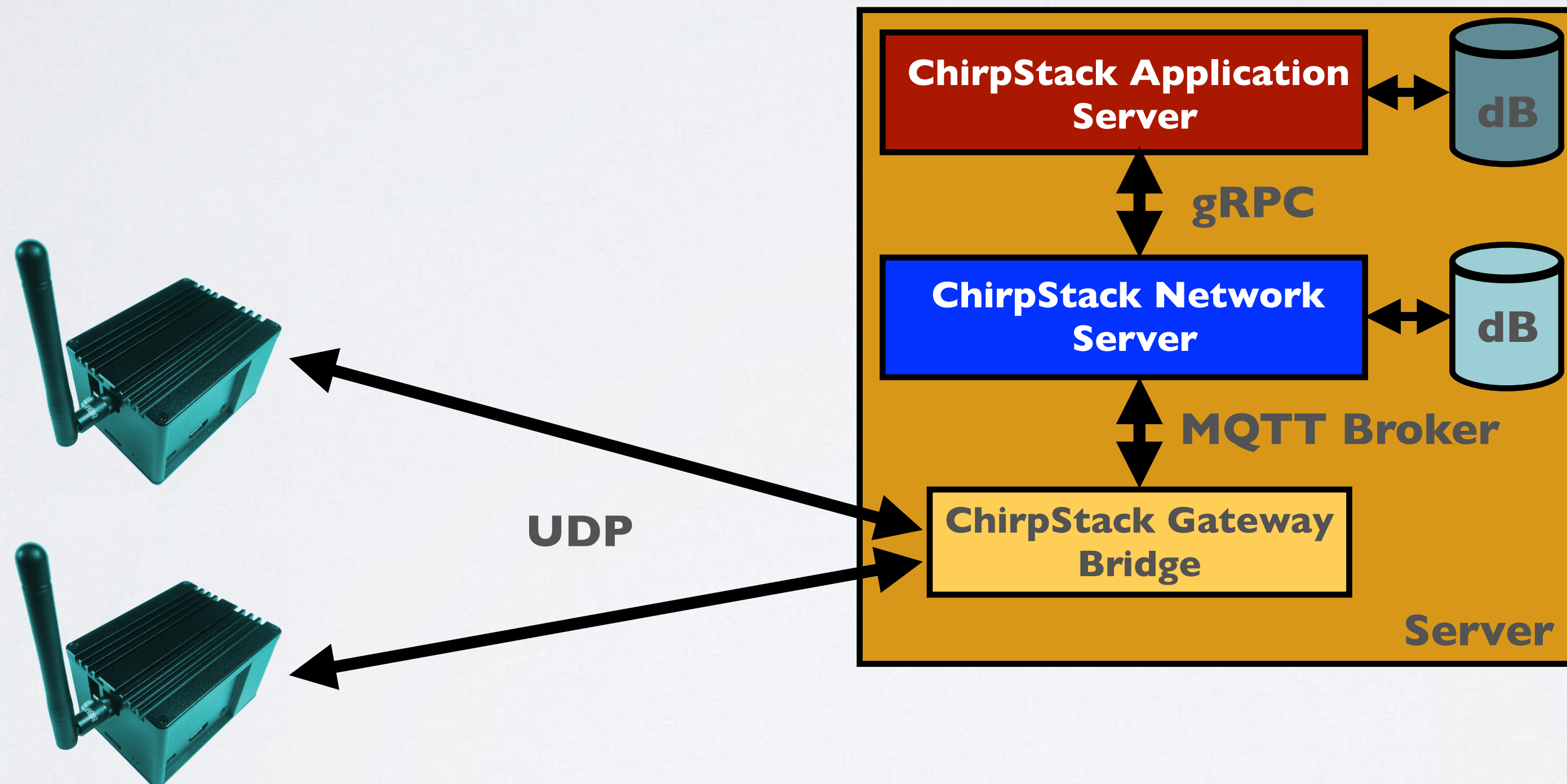
CHIRPSTACK COMPONENTS

- All ChirpStack components and their dependencies can be installed on the gateway itself, which I will demonstrate in this video.



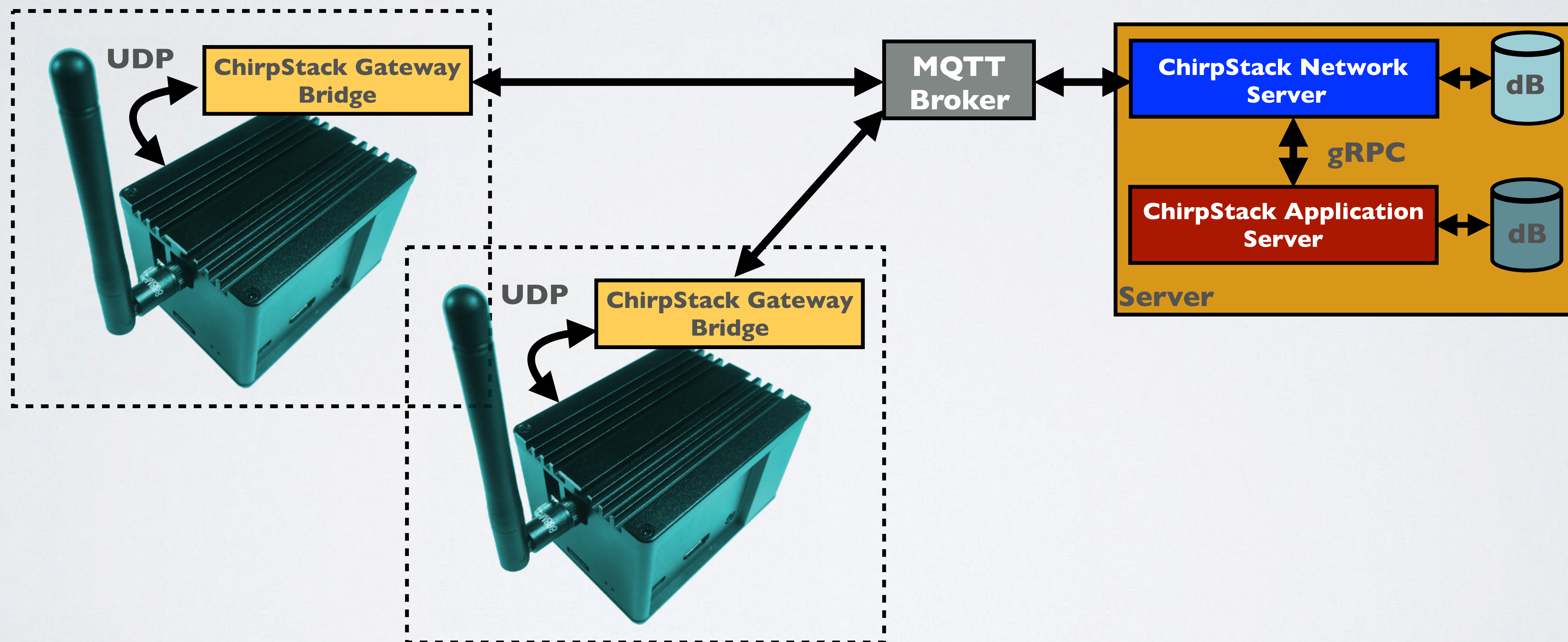
CHIRPSTACK COMPONENTS

- All ChirpStack components and their dependencies can be installed on a single server instance.



CHIRPSTACK COMPONENTS

- ChirpStack Gateway Bridge can be installed on the gateway itself, the other ChirpStack components and their dependencies can be installed on a single server



CHIRPSTACK ARCHITECTURE

Source:

<https://github.com/brocaar/loraserver>

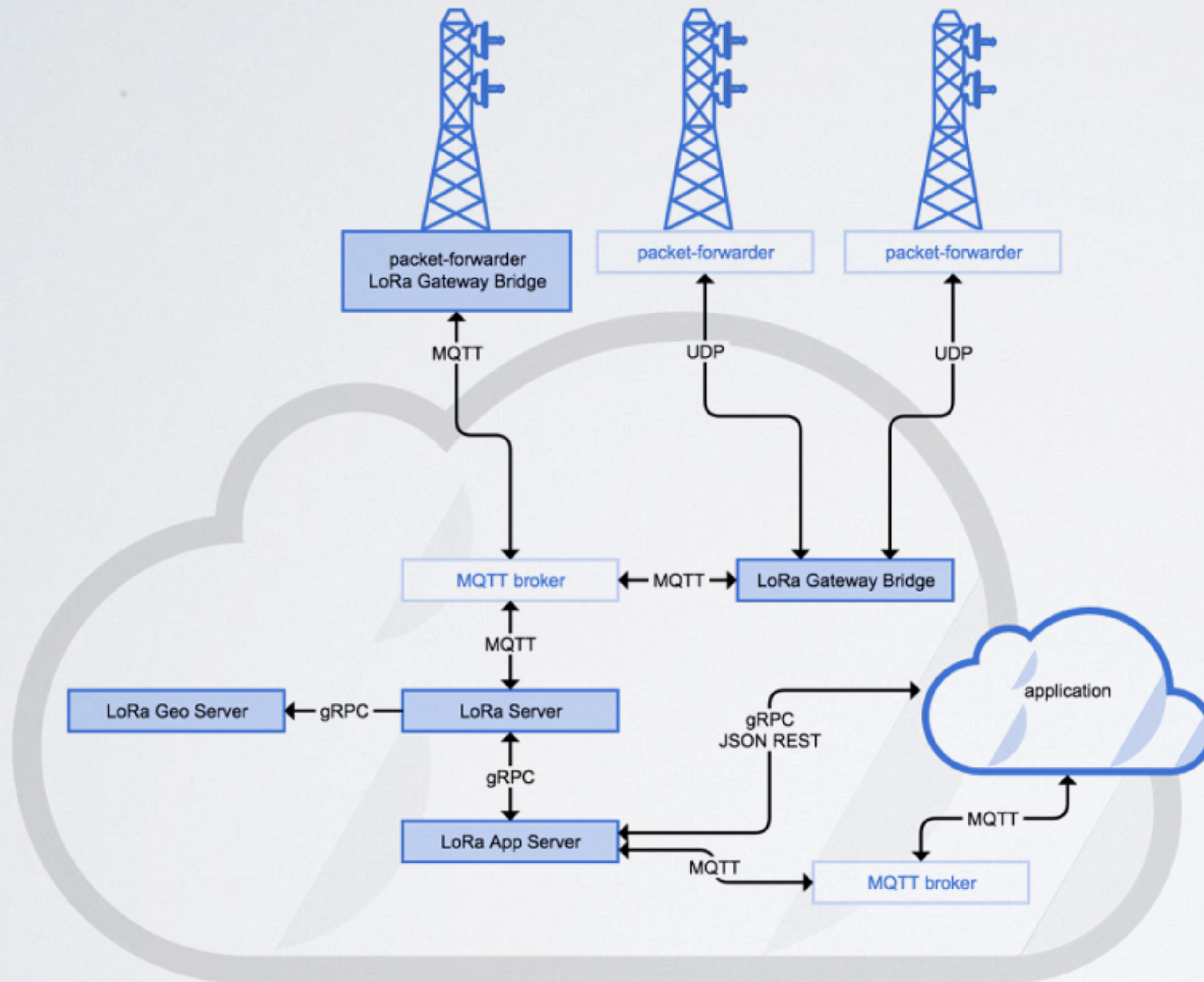
Note:

LoRa Gateway Bridge is renamed to ChirpStack Gateway Bridge

LoRa Geo Server is renamed to ChirpStack Geolocation Server

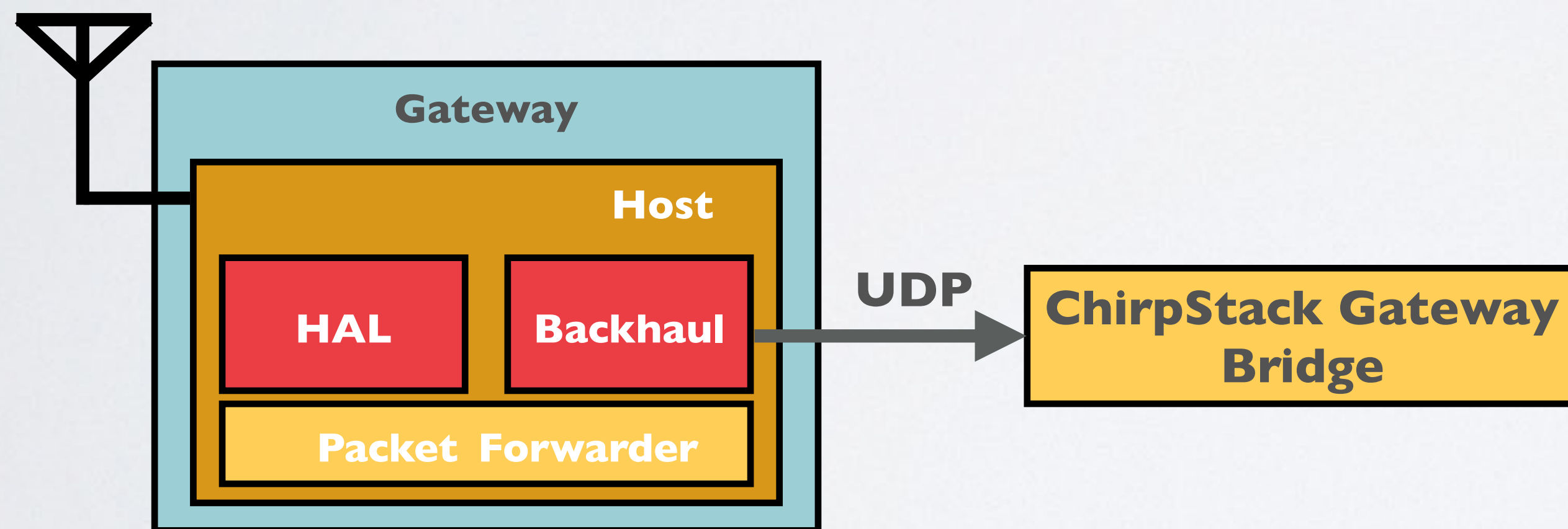
LoRa Server is renamed to ChirpStack Network Server

LoRa App Server is renamed to ChirpStack Application Server



HOW IT WORKS

- The gateway has the Semtech packet forwarder installed. The `global_conf.json` and `local_conf.json` files are used to configure the packet forwarder.
- In the `global_conf.json` file the ChirpStack Gateway Bridge server address and port (1700) are set. The packet forwarder has now all the information it needs to send data to the ChirpStack Gateway Bridge.



In this example the ChirpStack Gateway Bridge is installed on the Gateway.

`global_conf.json`

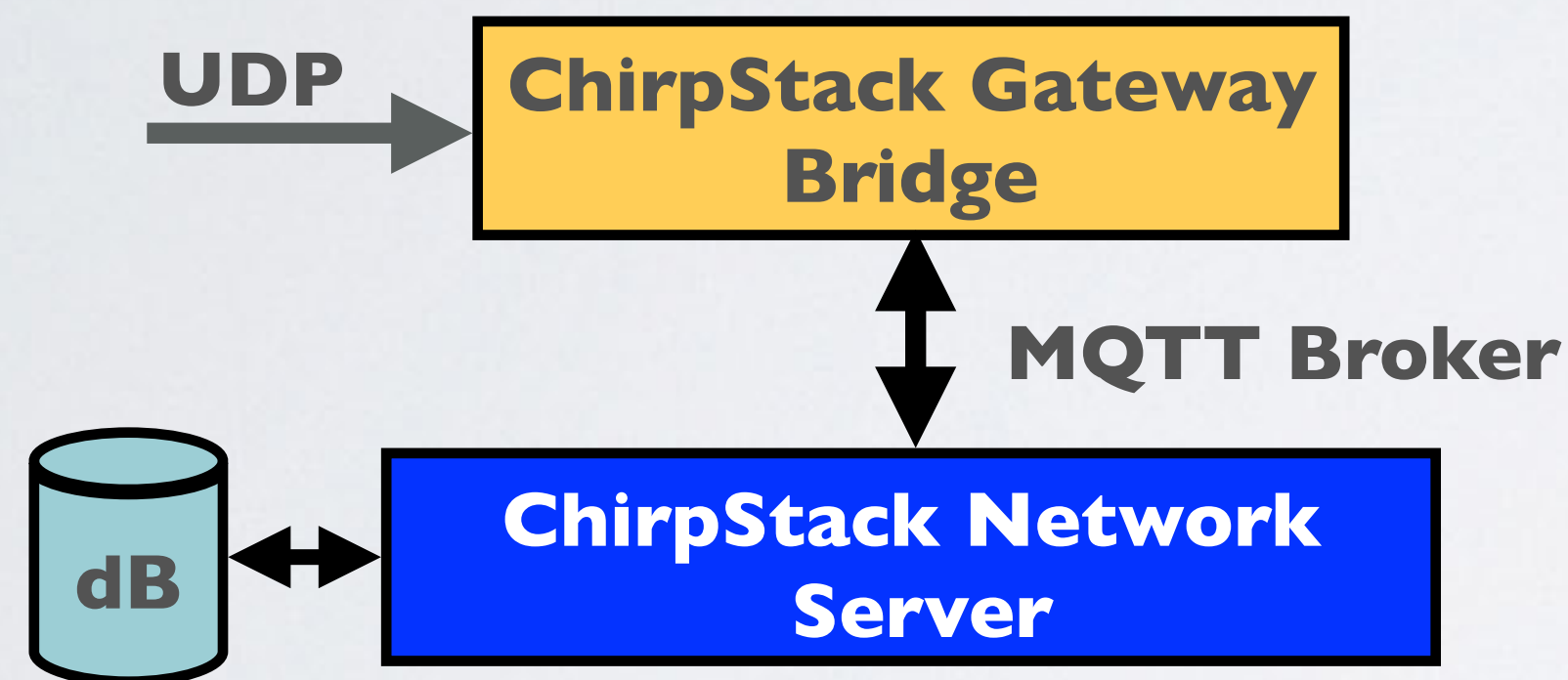
```
/* LoRa Gateway Bridge
server address and port
*/
"gateway_conf": {
  "server_address": "localhost",
  "serv_port_up": 1700,
  "serv_port_down": 1700
}
```

Note:

LoRa Gateway Bridge is renamed to ChirpStack Gateway Bridge

HOW IT WORKS

- The ChirpStack Gateway Bridge abstracts the received packet forwarder UDP protocol data. The data is converted into JSON and publishes it to a MQTT broker. The `chirpstack-gateway-bridge.toml` file is used to configure the ChirpStack Gateway Bridge.



`chirpstack-gateway-bridge.toml`

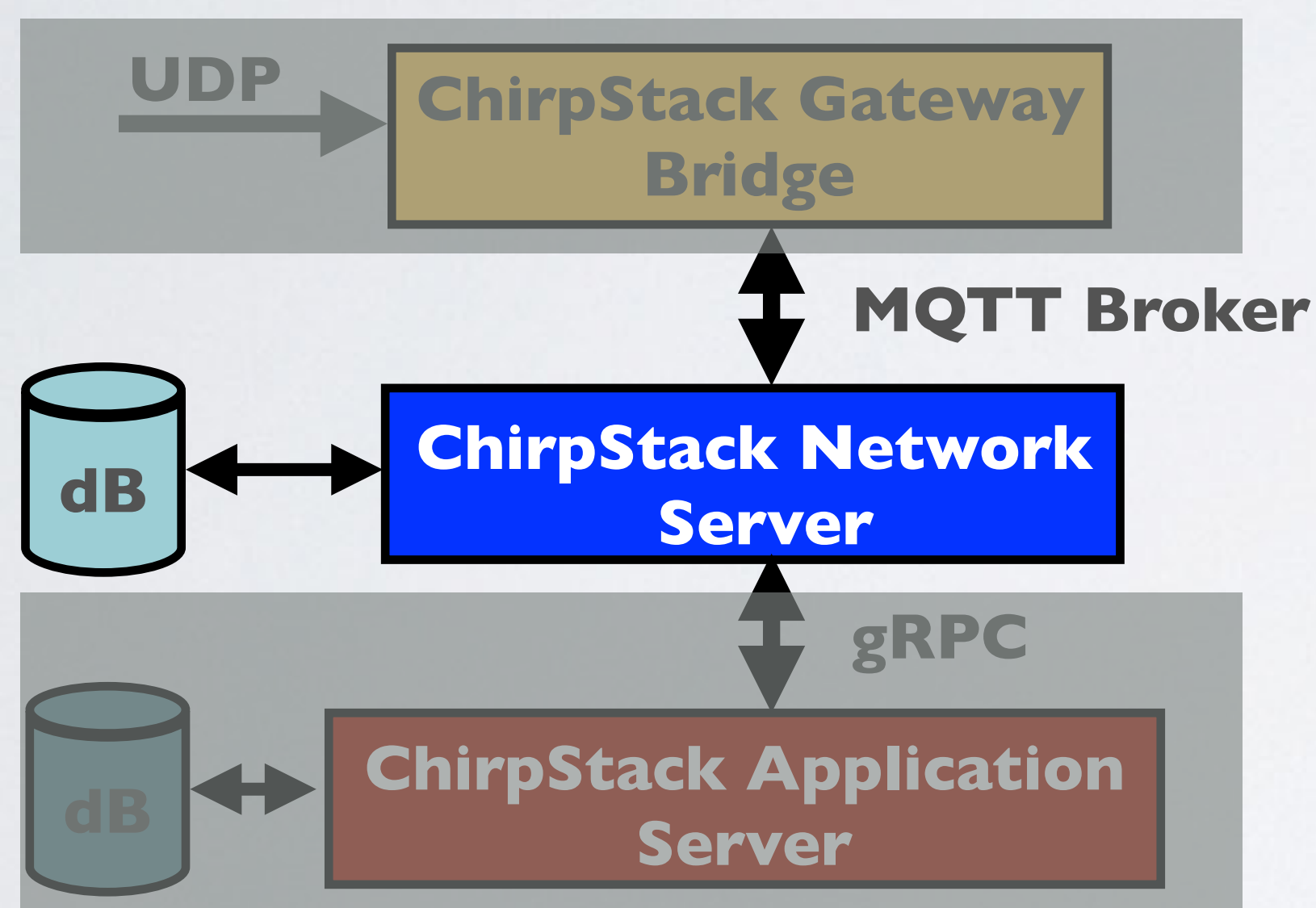
```
# MQTT broker address and port
server="tcp://127.0.0.1:1883"

# MQTT authentication
username=""
password=""
```

In this example the **MQTT Broker** and **ChirpStack Gateway Bridge** are installed on the same server.

HOW IT WORKS

- The ChirpStack Network Server is the LoRaWAN network server and it has a PostgreSQL database to store the gateway data. The `chirpstack-network-server.toml` file is used to configure the Chirpstack Network Server.



In this example the MQTT Broker, PostgreSQL database and ChirpStack Network Server are installed on the same server.

`chirpstack-network-server.toml`

```

# PostgreSQL settings
# user=loraserver_ns, password=dbpassword
# hostname=localhost, database=loraserver_ns
dsn="postgres://loraserver_ns:dbpassword@localhost/loraserver_ns?sslmode=disable"

#LoRaWAN regional band configuration.
name="EU_863_870"

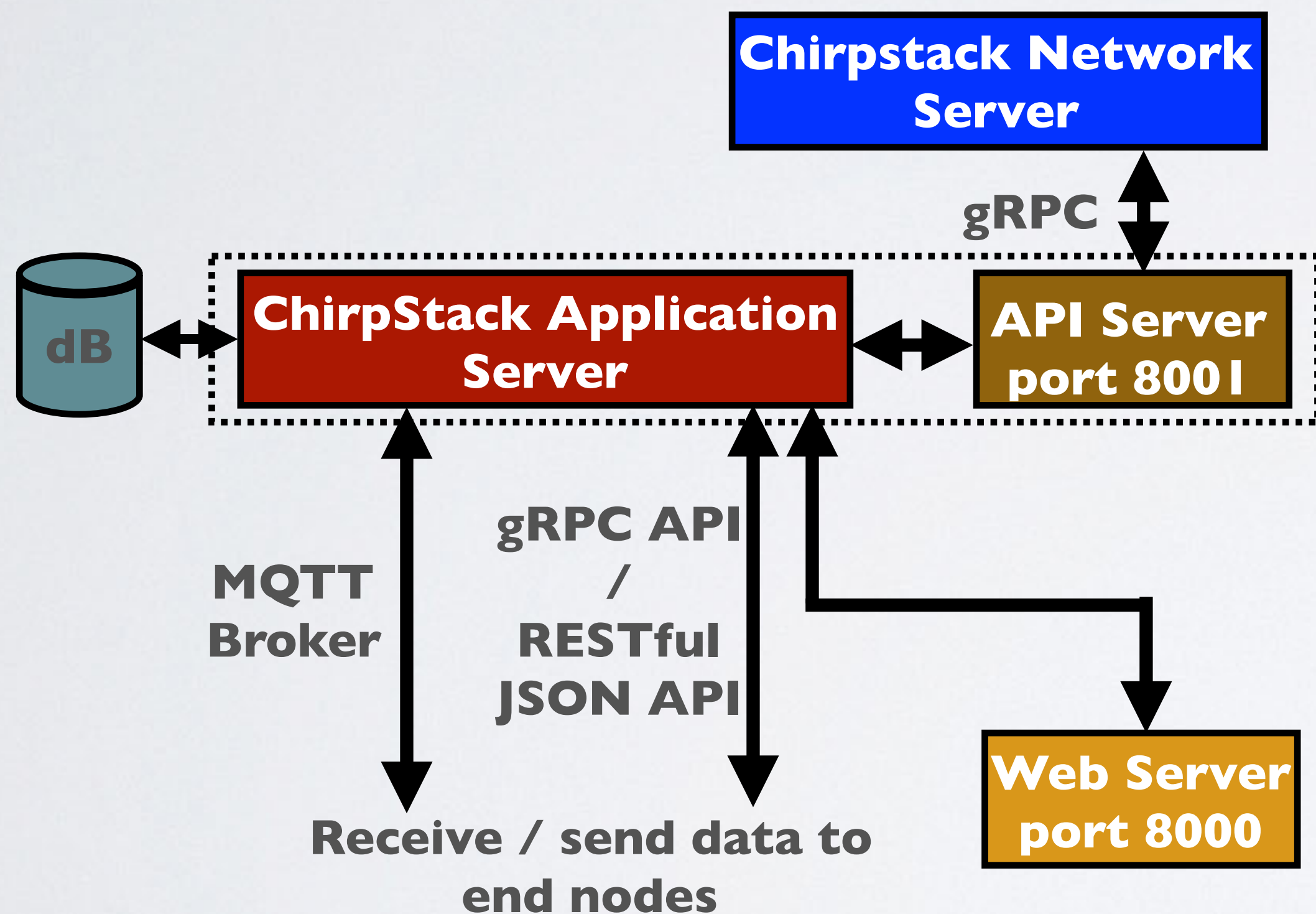
# MQTT broker address and port
server="tcp://localhost:1883"

# MQTT authentication
username=""
password=""

# Use the system's local time zone
timezone="Local"
  
```

HOW IT WORKS

- The ChirpStack Application Server is the LoRaWAN application server and it also has a PostgreSQL database to store the gateway data. The `chirpstack-application-server.toml` file is used to configure the ChirpStack Application Server.



`chirpstack-application-server.toml`

```
# PostgreSQL settings
# user=loraserver_as, password=dbpassword
# hostname=localhost, database=loraserver_as
dsn="postgres://loraserver_as:dbpassword@localhost/loraserver_as?sslmode=disable"

# MQTT broker address and port
server="tcp://localhost:1883"

# MQTT authentication
username=""
password=""

# The Internal API Server is used by LoRa Server
# to communicate with LoRa App Server
public_host="localhost:8001"
```

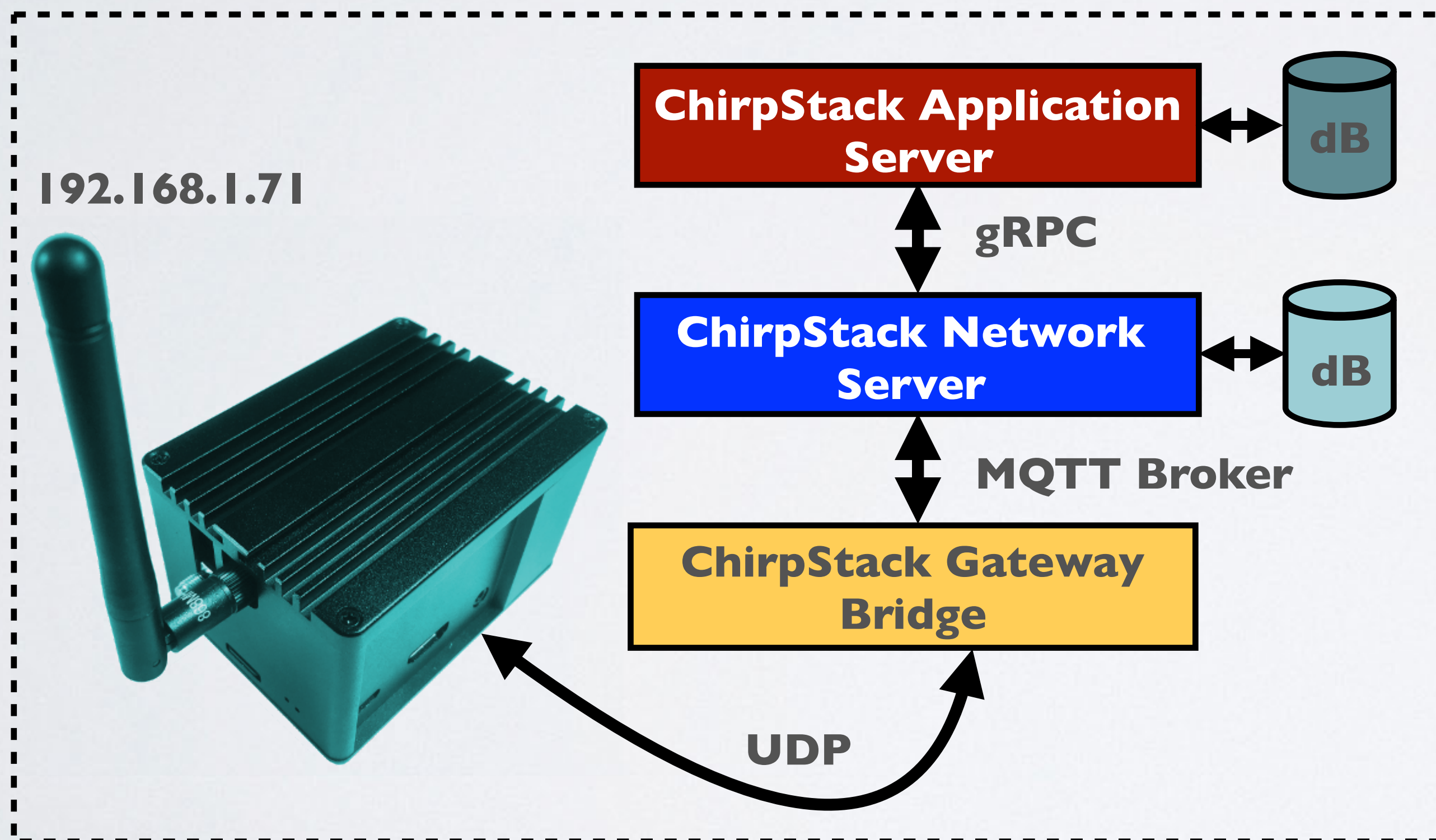

HOW IT WORKS

- This was a short and simplified explanation how the ChirpStack components works.
- The components have more configurations parameters which you can set.
For more information, read the documentation, see: <https://www.chirpstack.io>

USER MANUAL

- I have created a manual how to install and configure the ChirpStack Gateway Bridge, ChirpStack Network Server, ChirpStack Application Server and their dependencies on the RAK 831 Pilot Gateway.

https://www.mobilefish.com/developer/lorawan/lorawan_quickguide_chirpstack_udp_installation.html



ALL CHIRPSTACK COMPONENTS ON GATEWAY

Gateway

Configuration file: /opt/ttn-gateway/packet_forwarder/lora_pkt_fwd/global_conf.json

server_address=localhost (ChirpStack Gateway Bridge)

serv_port_up=1700 (the default port that ChirpStack Gateway Bridge is using)

serv_port_down=1700 (same)

ChirpStack Gateway Bridge

Configuration file: /etc/chirpstack-gateway-bridge/chirpstack-gateway-bridge.toml

MQTT broker address and port: server="tcp://127.0.0.1:1883"

ChirpStack Network Server

Configuration file: /etc/chirpstack-network-server/chirpstack-network-server.toml

PostgreSQL settings: dsn="postgres://chirpstack_ns:dbpassword@localhost/chirpstack_ns?sslmode=disable"

Automatically apply database migrations: automigrate=true

LoRaWAN regional band configuration: name="EU_863_870"

System's local time zone: timezone="Local"

MQTT broker address and port: server="tcp://127.0.0.1:1883"

ChirpStack Application Server

Configuration file: /etc/chirpstack-application-server/chirpstack-application-server.toml

PostgreSQL settings: dsn="postgres://chirpstack_as:dbpassword@localhost/chirpstack_as?sslmode=disable"

JWT secret: jwt_secret="zHj0RGotdJbgtKpPrv0/2axdER4bsnss6SHueANdEUg="

MQTT broker address and port: server="tcp://127.0.0.1:1883"

Internal API server: public_host="localhost:8001"

ALL CHIRPSTACK COMPONENTS ON GATEWAY

ChirpStack Application Server Web Interface

Web UI: [http\[s\]://<chirpstack_application_server_hostname_or_ip_address>:8080](http[s]://<chirpstack_application_server_hostname_or_ip_address>:8080)

Username: admin

Password: admin

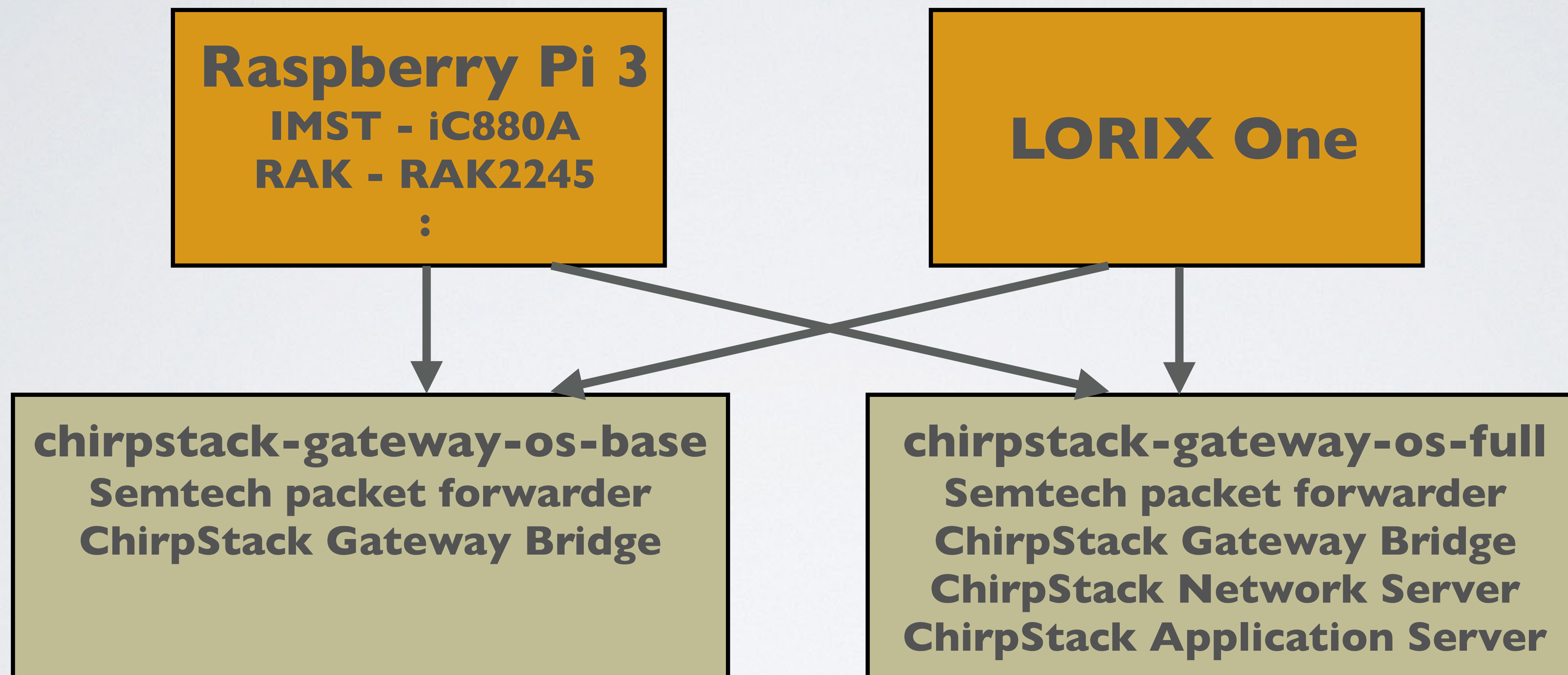
Chirpstack Application Server API

API: [http\[s\]://<chirpstack_application_server_hostname_or_ip_address>:8080/api](http[s]://<chirpstack_application_server_hostname_or_ip_address>:8080/api)

CHIRPSTACK GATEWAY OS

- The ChirpStack Gateway OS is an easy way to get started with LoRaWAN and the ChirpStack components.
- ChirpStack Gateway OS are SD card images which contains out-of-the-box support for the Semtech packet forwarder, ChirpStack Gateway Bridge, ChirpStack Network Server and ChirpStack Application Server for certain concentrator shields.
- More information: <https://www.chirpstack.io/gateway-os/overview/>

CHIRPSTACK GATEWAY OS



Status: May 2019

OTHER ALTERNATIVES

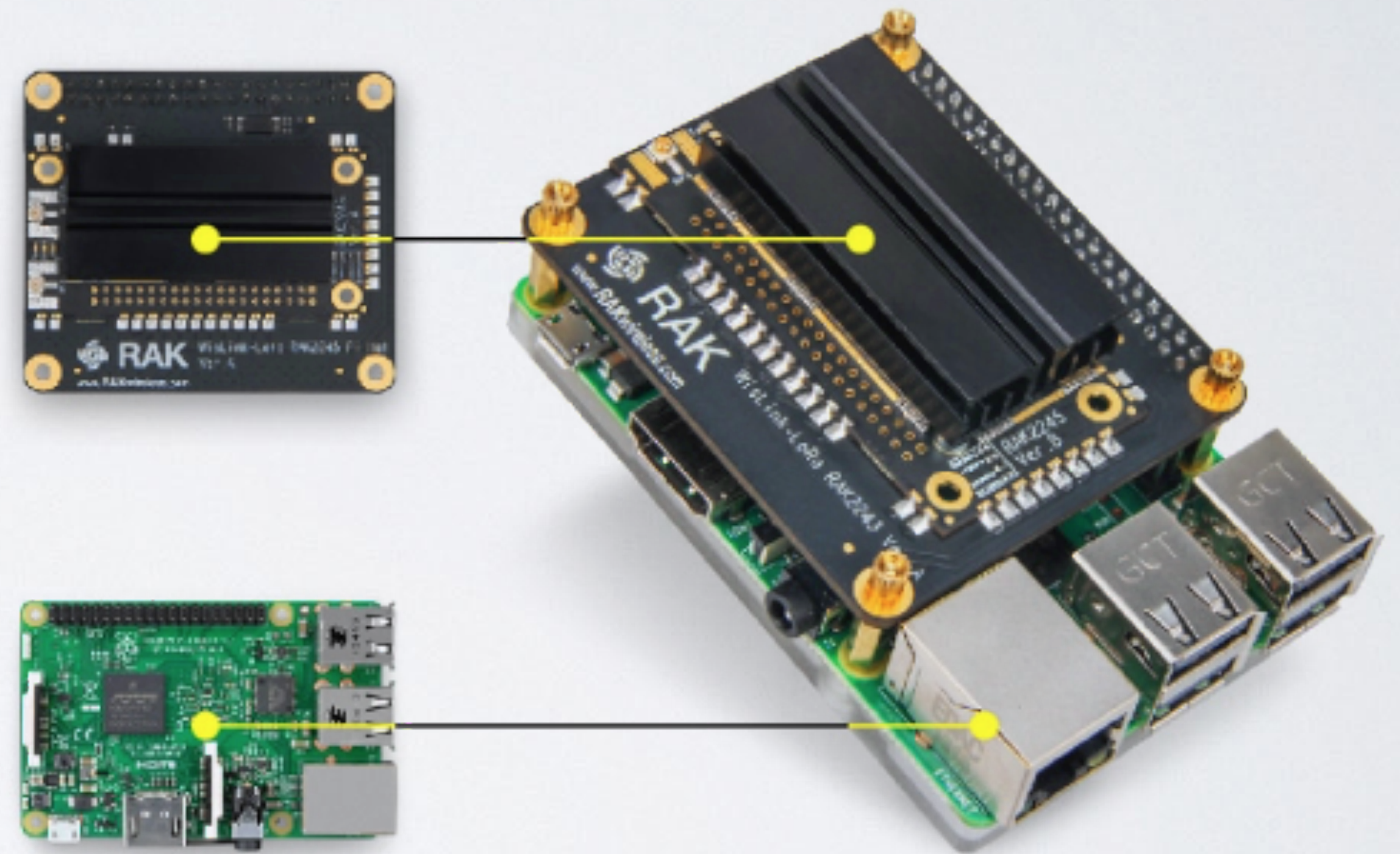
- There are also gateways on the market with the ChirpStack components already installed, such as the RAK WisKit.

This kit contains among other things:

- RAK2245 Pi HAT
(LoRa concentrator module)
 - Raspberry Pi 3B+
 - WisNode
(LoRa Node)
- A tutorial about this can be found at:
<https://www.hackster.io/fomi-T/simplest-lora-starter-kit-w-rak2245-rpi-ttn-loraserver-0ad993>



RAK2245 Pi Hat + Raspberry Pi
Prototype To Commercial



SD CARD WEAR OUT

- An SD Card **MAY** wear out. The lifetime of SD cards is limited by the number of writes.
- It is recommended NOT to use cheap SD cards.
- ChirpStack tries to minimise the number of database writes.