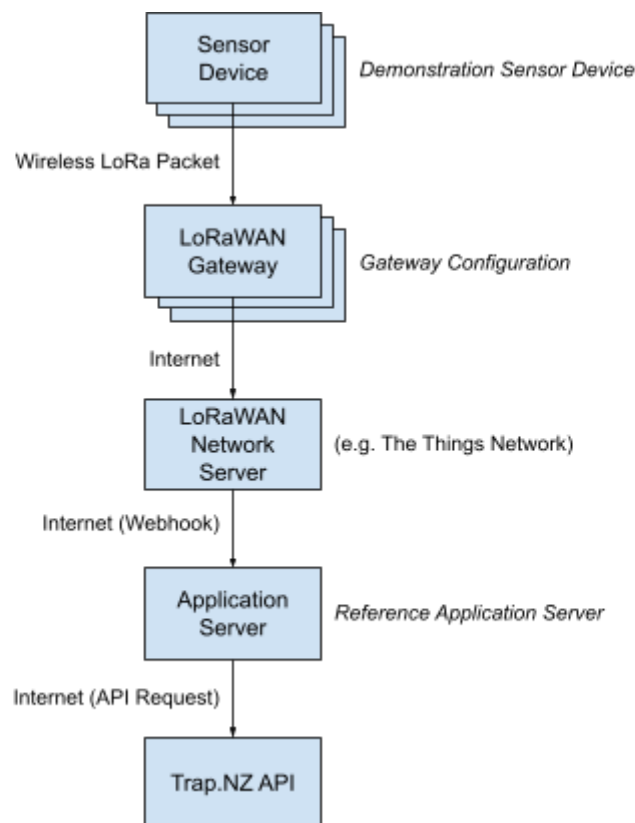# Integrating LoRaWAN sensors with Trap.NZ

This is a blueprint for projects to use to affordably configure sensors to submit data to Trap.NZ using LoRaWAN, a low power long range wide area network protocol.

We provide three components. A *gateway configuration* document explains how to take an off-the-shelf gateway device and set it up to provide LoRaWAN network coverage to nearby sensor devices. A *demonstration sensor device* provides a proof of concept end device, which senses and reports the status of a trap. A *reference application server* receives messages from a sensor via a LoRaWAN network server and relays those messages on to Trap.NZ.

Gateway Configuration

Our gateway configuration targets the off-the-shelf Mikrotik wAP LR9 kit. It describes how to configure the device to act as a LoRaWAN gateway for The Things Network.

Demonstration Sensor Device

Our demonstration sensor device is based on the common TTGO T-Beam hardware from LILYGO. It is assembled with a 0.96 inch OLED display, an 18650 Li-ion battery, and a switch between two pins to sense trap status.

Pictured is an assembled demo device. The OLED display is visible on the left of the board. The LoRa antenna attaches to the top. The battery is attached underneath the board (not visible). The blue wire at the bottom of the board represents the switch to sense trap status.

The software and documentation for this device can be found at https://github.com/Groundtruth/sensor-trap-demo-device.

Once assembled and programmed, the device will connect to a LoRaWAN network and transmit status information both periodically and on status-change events. It implements some basic power-saving measures and reports its battery status. When a debug button is pressed, the OLED displays some diagnostic information about the state of the device. The software can be configured for varying LoRaWAN connection parameters, and various sensor message timings.

Particular attention has been paid to the logic around sending heartbeat messages and safely recording the trap's status. For the case of live traps, it is vital that a sensor never report "set" (open) when it is actually "sprung" (closed).

This hardware is easy to procure and assemble, and the software demonstrates the required functions for a sensor trap. However, the work here was limited in scope and it should be noted that it should not be deployed without further thought. Future work to produce a deployable sensor would involve optimising for lower cost, longer battery life, and performing testing to establish reliability in the field. The current state of this work should provide a starting point for this future work.

Reference Application Server

The reference application server receives messages from sensor devices, forwarded by a LoRaWAN network server such as The Things Network. It uses client code generated from the Trap.NZ OpenAPI spec to create sensor records on Trap.NZ corresponding to incoming sensor messages.

The software and documentation for the reference application server can be found at
https://github.com/Groundtruth/sensor-trap-reference-application.

The reference application server is written in Typescript with minimal dependencies. It opens an HTTP server to receive messages from a The Things Network webhook. The device of an incoming message is uniquely identified by its LoRaWAN DevEUI (device extended unique identifier), which is used to look up corresponding device parameters such as its Trap.NZ sensor ID and expected timeout. An appropriate sensor message is constructed and sent to the Trap.NZ API. The software produces log messages which can be used to monitor and alert for error conditions.

The software can be configured with authorization details for a Trap.NZ sensor provider account. Device configuration is read dynamically from a JSON file, so device information can be updated without restarting the software. Other functions of the software (e.g. webhook listener, device information lookup, sensor message format) are written so as to be easily customised by a sensor provider.

The software is published so that a sensor provider (a person or organisation producing and supporting sensor devices) could adapt and run it for their own devices, or look to it for inspiration on how to implement their own application server.