```python
import sys
import json
import paho.mqtt.client

# Subscriber
APPLICATION_ID = "centraltcc"
TOPIC = "centraltcc/devices/sinknode/up"
HOSTNAME = "brazil.thethings.network"
PORT = 1883
TIMEOUT = 60
client = paho.mqtt.client.Client(client_id='ttn_client', clean_session=False)

# Publisher
WEBSOCKET_HOST = "broker.hivemq.com"
WEBSOCKET_PORT = 8000
WEBSOCKET_TOPIC = "vagas/cpd/"
websocket_client = paho.mqtt.client.Client(client_id='websocket_client', transport="websockets")

def on_connect(client, userdata, flags, rc):
  client.subscribe(topic=TOPIC, qos=2)

def send_data_to_app(data="hello world"):
    websocket_client.publish(WEBSOCKET_TOPIC, payload=data)

def on_message(client, userdata, message):
  # Obtém dados da mensagem
    PAYLOAD = message.payload

  # Decodifica mensagem e envia para a aplicação
    data = json.loads(PAYLOAD.decode("utf-8"))["payload_raw"]

    send_data_to_app(data)

if __name__ == '__main__':
  # Cria e configura cliente que obtém dados da TTN
  client.username_pw_set(APPLICATION_ID, ACCESS_KEY)
  client.on_connect = on_connect
  client.on_message = on_message
  client.connect(HOSTNAME, PORT, TIMEOUT)

  # Cria e configura cliente mqtt para enviar dados ao front end
  websocket_client.connect(WEBSOCKET_HOST, WEBSOCKET_PORT, TIMEOUT)

  # Inicia clientes
  websocket_client.loop_start()
  client.loop_start()

  # Aguarda mensagens
  while True:
    pass

  # Encerra clientes
  client.loop_stop()
  client.disconnect()
  websocket_client.loop_stop()
  websocket_client.disconnect()
  sys.exit(0)


```