

Hi It's Marvin Hosea

- Software Engineer @Scaratec Gmbh
- I have been involved in the integration and implementation of Integrated Customs Management System (iCMS) for Sea and Air import championed by Kenya Revenue Authority (KRA)
- Gophers Africa And Nairobi Gophers Co-organizer

Saving Africa Savannah Wildlife With Go MQTT

GopherCon 9 July 2024

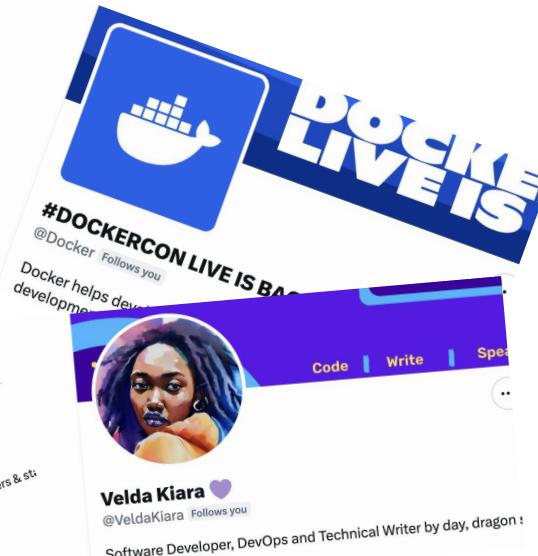
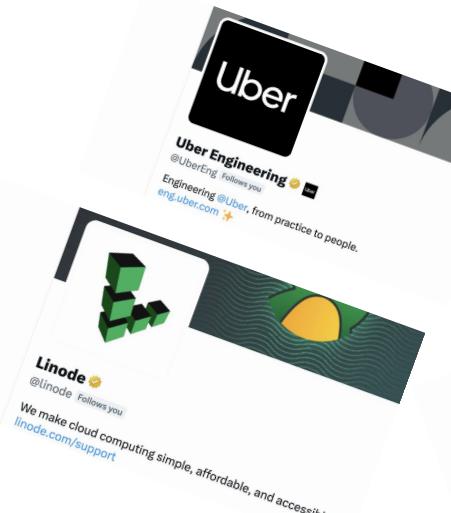
Let's Manage Expectations

- **What I will talk about**

- The Elephant in The Room
- Go MQTT Overview
- Challenges and Implementation
- Closing Remarks

Questions?

@marvin_hosea



Velda Kiara
@VeldaKiara Follows you
Software Developer, DevOps and Technical Writer by day, dragon by night!

Intro

The Elephant in The Room

Wildlife Poaching

Illegal trafficking and killing of animals

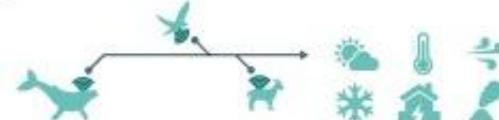
Possible now



(A) Temporary conservation action based on prediction of animal movement



(B) Cars, doorbell cameras, edge computing



(C) Animals as sensors (buoys)



(D) Coordinated disease monitoring

Futuristic



Change in activity of animals to alert to presence of poachers in an area



Counting insects with cars



Early earthquake and volcanic detection



One health: combine domestic, wild, human

Kenya Stats

Source KWS

- 1970 - 1980s: 170,000 Elephants

- 1979 - 2000: 16,000 Elephants | Ivory Demand



100,000 elephants killed in Africa, study finds

Posted on agosto, 21 2014

By Jason Straziuso Associated Press NAIROBI, Kenya —

Poachers killed an estimated 100,000 elephants across Africa between 2010 and 2012, a huge spike in the continent's death rate of the world's largest mammals because of an increased demand for ivory in China and other Asian nations, a new study published Monday found.

Warnings about massive elephant slaughters have been ringing for years, but Monday's study is the first to scientifically quantify the number of deaths across the continent by measuring deaths in one closely monitored park in Kenya and using other published data to extrapolate fatality tolls across the continent.



The elephant population in Tanzania's Selous Game Reserve dropped from 40,000 to 13,000 over the last three years.

© Associated Press

Impact

- Formation of Kenya Wildlife Service
- Africa Elephant is listed in Appendix I
- World Elephant Day: 12th August
- World Anti-poaching Systems and Strategies Improvement



Anti-poaching

Anti-poaching strategies

- Patrols and Rangers
 - Provide necessary training, and
- Equipment for locals to combat poaching



Source
MEPs

Anti-poaching strategies

- Patrols and Rangers
 - Provide necessary training, and Equipment for locals to combat poaching
- Surveillance
 - Cameras to monitor wildlife and detect poachers

Kenya Stats

Source KWS

- 1970 - 1980s: 170,000 Elephants
- 1979 - 1990: 16,000 Elephants | Ivory Demand

Current population: Approx **36,000** Elephants

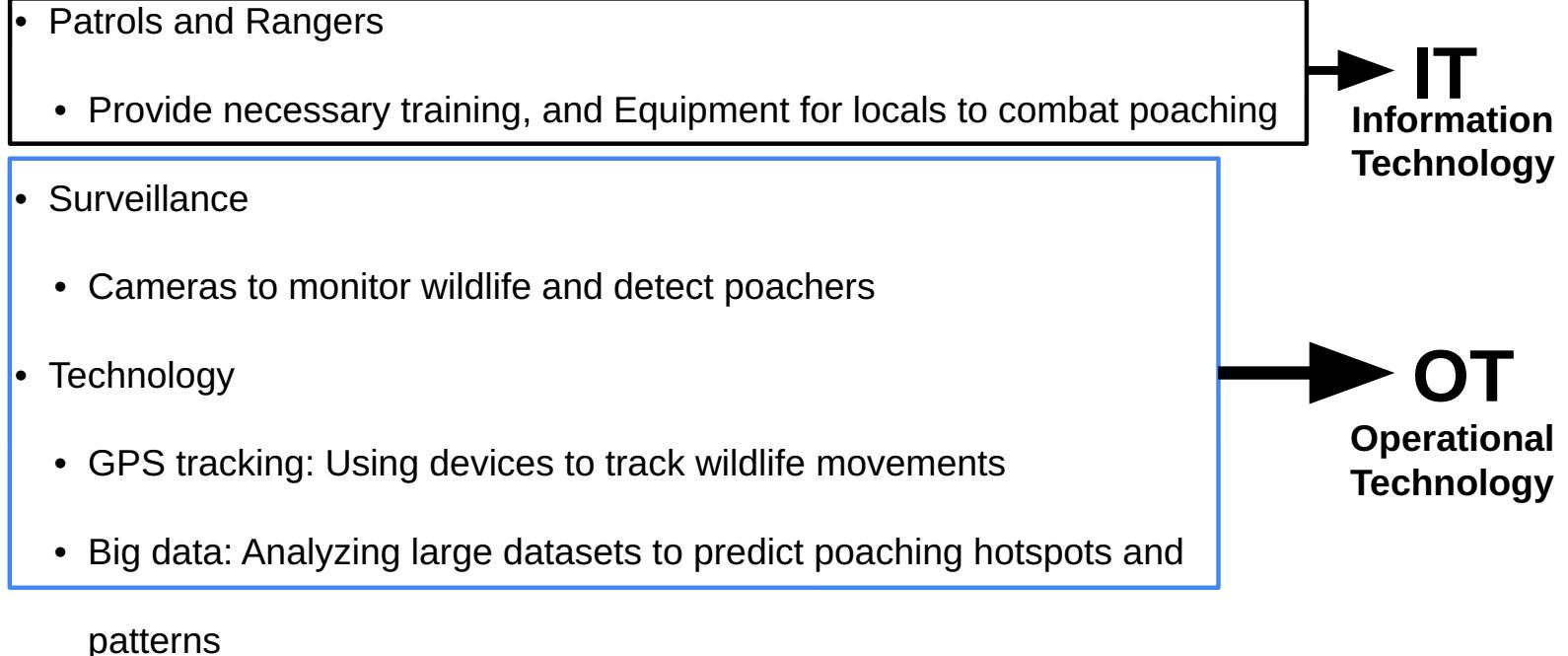
36,000

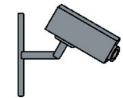


technology

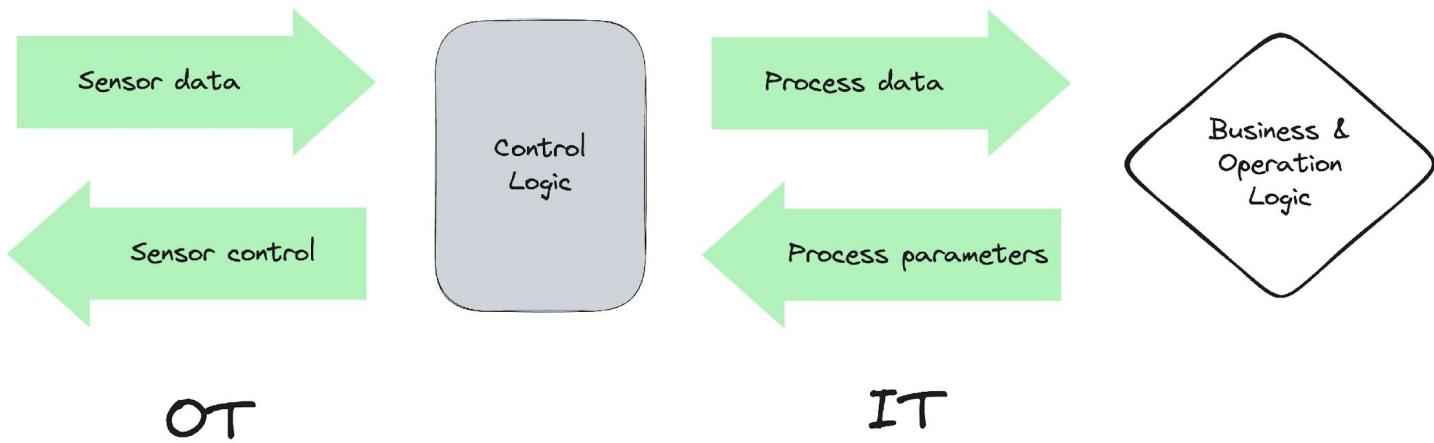
Anti-poaching strategies

- Patrols and Rangers
 - Provide necessary training, and Equipment for locals to combat poaching
- Surveillance
 - Cameras to monitor wildlife and detect poachers
- Technology
 - GPS tracking: Using devices to track wildlife movements with Geo fencing
 - Big data: Analyzing large datasets to predict poaching hotspots and patterns





Devices



OT

IT

- Patrols and Rangers

- Provide necessary training, and equipment for locals to combat poaching.

- Surveillance

- Cameras to monitor wildlife and detect poachers

- Technology

- GPS tracking: Using devices to track wildlife movements

- Big data: Analyzing large datasets to predict poaching hotspots and

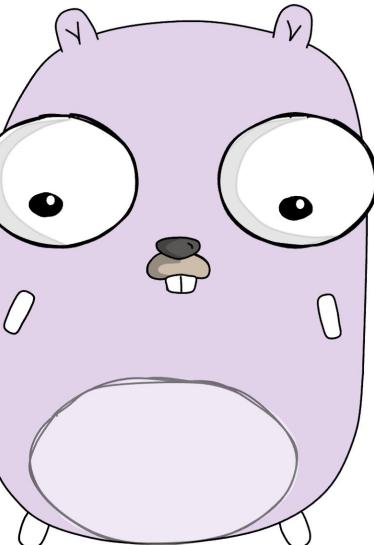
patterns

IT
Information
Technology

OT
Operational
Technology

Challenge

- Expensive cost of internet bundles
- Unreliable internet within the Savannah region
- Reduce cost of anti-poaching operations



Can We Talk About Go Now

Choosing The Right Technology For The Job

System Requirement

- Low bandwidth and high latency environments
- Distribute minimal data in huge volumes

Work with constrained devices (**microcontrollers**)

- Highly scalable
- Secure
- Use Go



WHAT IS MQTT?

Message Queuing Telemetry Support

MQTT is a lightweight, publish-subscribe based **binary** messaging protocol designed for resource-constrained devices and low-bandwidth, high-latency, or unreliable networks.

- Minimal resources consumption
- Low bandwidth and high-latency environment
- Bi-directional
- Easy to implement: 15 protocol command
- Has a fixed 2 byte over-the-wire overhead header
- Has a max 256MB payload
- Runs on top of TCP/TLS/WS/WSS stacks

● mqtt
Search term

+ Compare

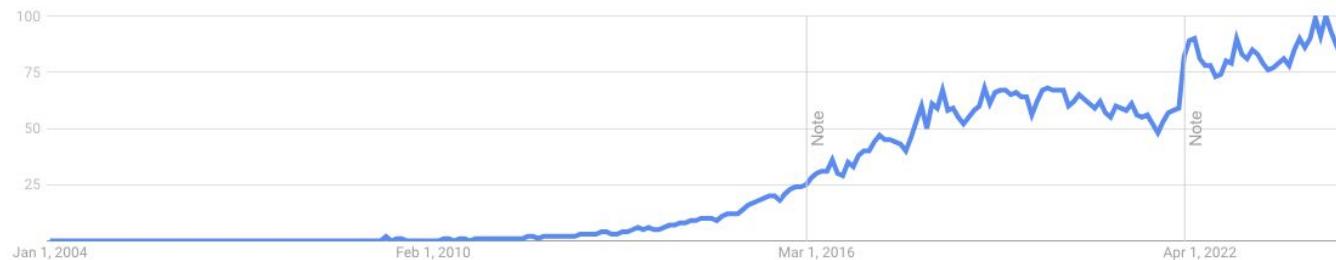
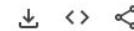
Worldwide ▾

2004 - present ▾

All categories ▾

Web Search ▾

Interest over time ?





Internet Of Things



Internet Of Animals

Made with Piñata Farms

Application of MQTT

- Remote sensing
- Smart Cities/Homes
- Wearables
- Car Monitoring
- Livestock management

MQTT Flow

MQTT Flow

Client initiate a **connection** to the broker

MQTT Flow

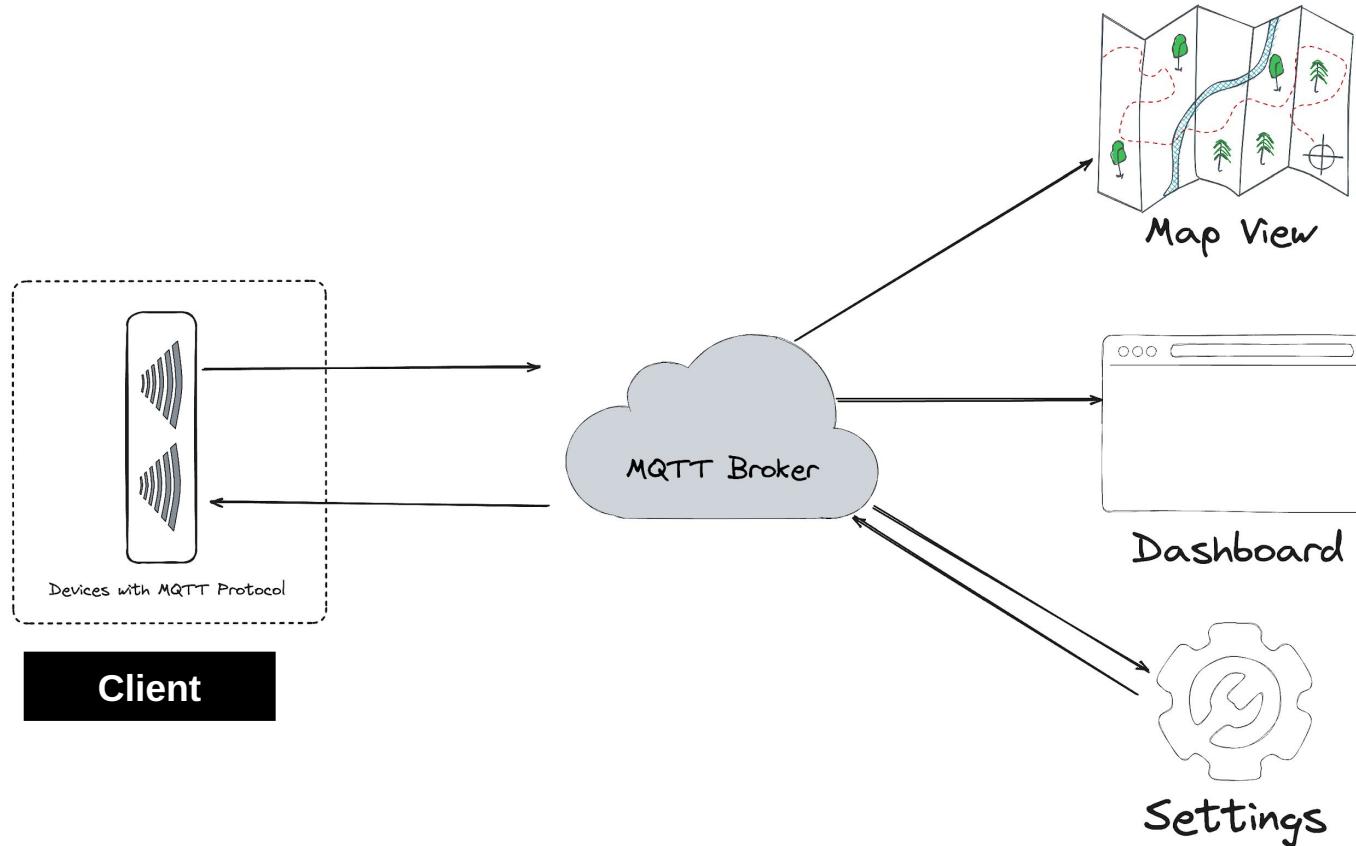
1. Client initiate a connection to the broker

Client either **subscribe** or **publish** a message to a topic

MQTT Flow

1. Client initiate a connection to the broker
2. Client either subscribe or publish a message to a topic

Broker receives published message and forward them to all clients subscribed to relevant topics



Go

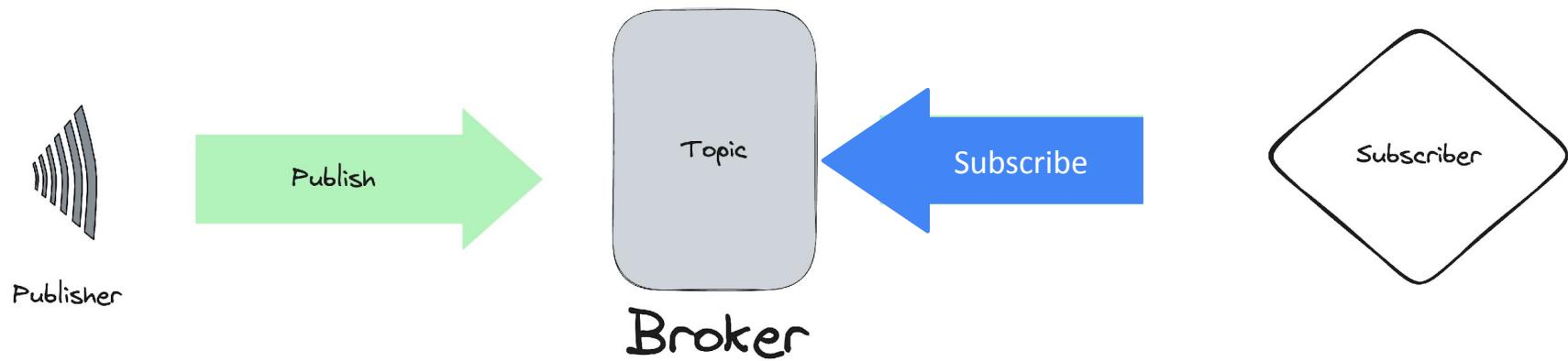


Go and MQTT Powerful Alliance

- Go's efficiency and low resource requirements make it ideal for resource-constrained devices.

Implementation

Simple Architecture



Go And MQTT Prepare a Broker

- Mosquitto



Free Public MQTT Broker

Others

- Emitter
- EMQ X
- HiveMQ
- AWS IoT Core

*"Some dev have no problem
making public test server their
production"*



Kenya Stats

Source KWS

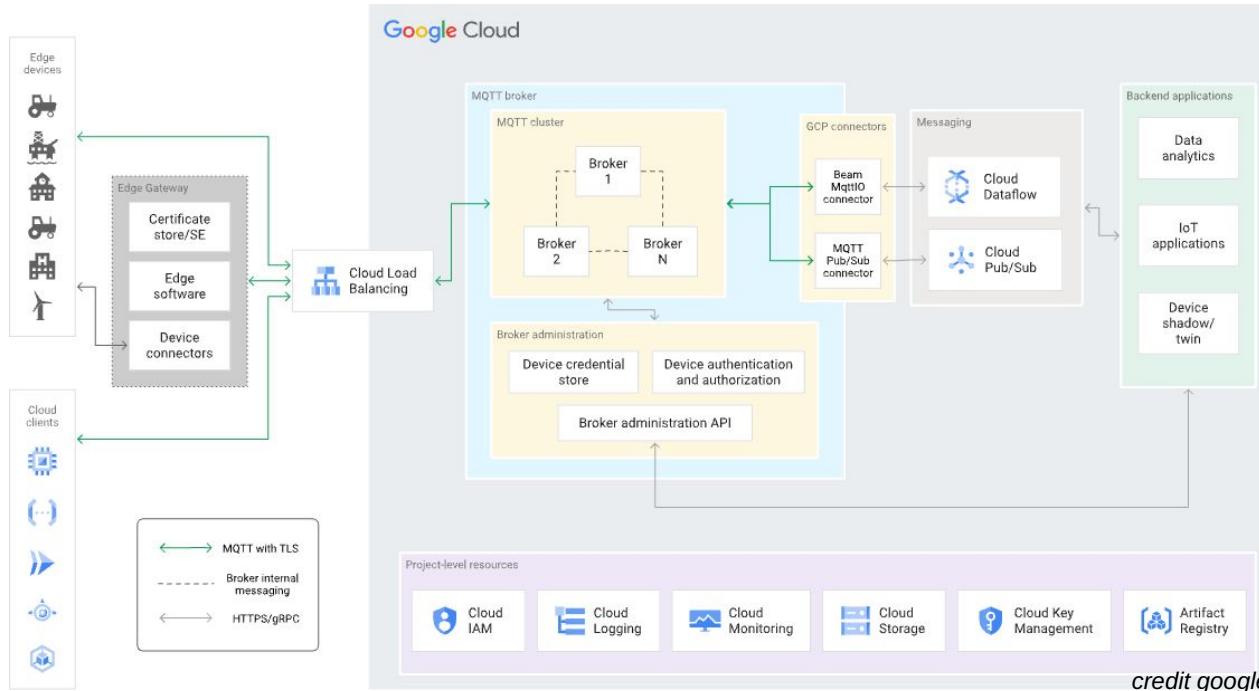
- 1970 - 1980s: 170,000 Elephants
 - 1979 - 1990: 16,000 Elephants | Ivory Demand
- Current population: Approx **36,000** Elephants and other endangered species

36,000



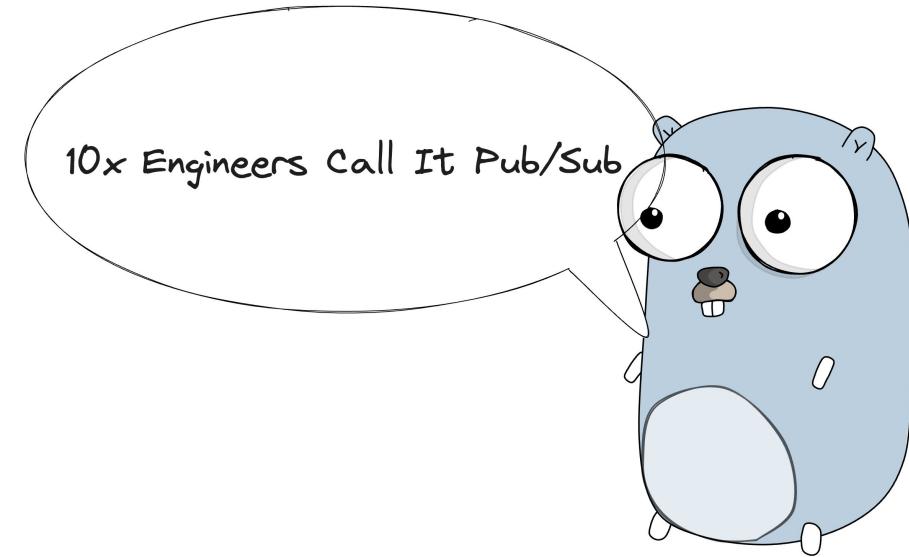


HA MQTT Architecture



Publish/Subscribe Pattern

Publisher and Subscriber



Go and MQTT

Go MQTT Library

github.com/eclipse/paho.mqtt.golang MQTT v3

github.com/eclipse/paho.golang MQTT v5

Code



Publisher Example

```
1 ...
2
3 client := mqtt.NewClient(opts)
4 token := client.Connect()
5 if token.Wait() && token.Error() != nil {
6     log.Fatal(fmt.Sprintf("Connection failure: %v", token.Error()))
7 }
8
9 for i := 1; i <= 10; i++ {
10    message := fmt.Sprintf("Happy Gophercon, Happy Gopher %d", i)
11    token := client.Publish(cfg.Topic, 0, false, message)
12    token.Wait()
13
14    time.Sleep(1 * time.Second)
15 }
16
17 ...
```



Subscriber Example

```
● ● ●  
1 .....  
2  
3 client := mqtt.NewClient(opts)  
4 token := client.Connect()  
5 if token.Wait() && token.Error() != nil {  
6     log.Fatal(fmt.Sprintf("Connection failure: %v", token.Error()))  
7 }  
8  
9 token = client.Subscribe(cfg.Topic, 0, func(client mqtt.Client, msg mqtt.Message) {  
10    fmt.Println("Sub1: Received message:", string(msg.Payload()))  
11 })  
12 if token.Wait() && token.Error() != nil {  
13     log.Fatal(fmt.Sprintf("Subscription failure: %v", token.Error()))  
14 }  
15  
16 .....
```



TOO EASY

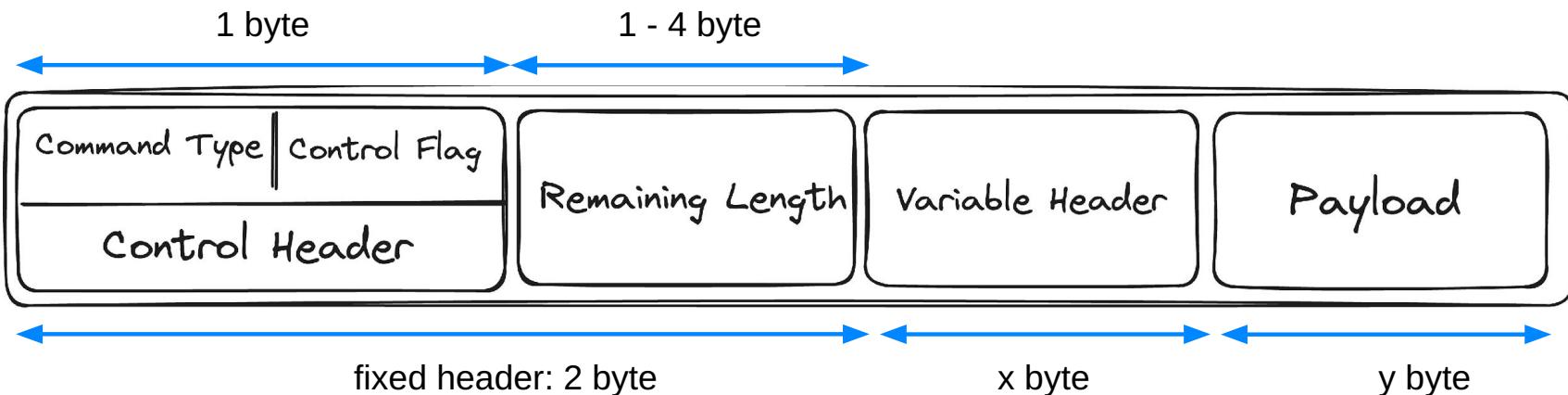
Made with Piñata Farms

Disappointed!

Paho!

- Indefinite I/O block
- Failure to handle continues reconnection
- Uses `any` for the payload
- Excessively asynchronous API
- **Existing packages are hefty for most microcontrollers**

Let's Take a look at the MQTT packet



Let's Take a look at the MQTT packet

Fixed header: Made up of the packet type, packet flags, and length of the remaining packet.

Let's Take a look at the MQTT packet

Variable header: The content and length of the variable header will vary according to the packet type and includes packet-type specific meta-data

Let's Take a look at the MQTT packet

Payload: The packet-type specific payload data

Simple Go implementation of CONNECT command



TCP connection to the MQTT broker

```
1 .....
2
3 conn, err := net.Dial("tcp", "test.mosquitto.org:1883")
4 if err != nil {
5     log.Fatal(err)
6 }
7
8 .....
```



Constructs the MQTT CONNECT packet and sends it to the broker.

```
1 .....
2
3 packet := []byte{
4     1 << 4, 0, // Fixed header
5     0, 4, 'M', 'Q', 'T', 'T', // Protocol name
6     5, // Protocol version
7     2, // Connect flags (Clean Session)
8     byte(60 >> 8), byte(60), // Keep alive
9     0, // Properties length
10 }
11
12 clientID := "gophercon_2024"
13 clientIDLength := len(clientID)
14 packet = append(packet, byte(clientIDLength>>8), byte(clientIDLength))
15 packet = append(packet, []byte(clientID)...)
16
17 packet[1] = byte(len(packet) - 2)
18
19 _, err = conn.Write(packet)
20 if err != nil {
21     log.Fatal(err)
22 }
23
24 .....
```

Reads the CONNACK packet from the broker.

```
1 .....
2
3 connack := make([]byte, 4)
4 _, err = conn.Read(connack)
5 if err != nil {
6     log.Fatal(err)
7 }
8
9 if connack[3] != 0 {
10    log.Fatal("connection refused")
11 }
12
13 fmt.Println("Connected successfully")
14
15 // implement the rest {keep-alive and reconnection etc} of the strategies
16 select {}
17
18 .....
```



Many moons and cup of coffees
later...
Demo

```
→ mqtt code _  
→ mqtt app-mqtt  
→ app-mqtt teller run npx cypress run --env allure=true  
  
→ app-mqtt teller run air cmd/subscriber/main.go  
-- teller: loaded variables for app-mqtt using .teller.yml --  
FATA[0000] could not execute command error="exec: \"air\": executable file not fo  
und in $PATH"  
→ app-mqtt teller run air cmd/subscriber/main.go  
-- teller: loaded variables for app-mqtt using .teller.yml --  
FATA[0000] could not execute command error="exec: \"air\": executable file not fo  
und in $PATH"  
→ app-mqtt export $(egrep -v '^#' .env | xargs) && go run cmd/subscriber/main.go  
# gophercon/app-mqtt  
mqtt/client.go:245:4: c.FixedHeader undefined (type *Client has no field or method FixedHeader)  
mqtt/client.go:245:26: c.Type undefined (type *Client has no field or method Type)  
mqtt/client.go:245:36: undefined: f  
→ app-mqtt gld _  
→ app-mqtt gld _  
→ app-mqtt export $(egrep -v '^#' .env | xargs) && go run cmd/publisher/main.go  
Published: Publishing message 1  
Published: Publishing message 2  
Published: Publishing message 3  
Published: Publishing message 4  
Published: Publishing message 5  
Published: Publishing message 6  
Published: Publishing message 7  
Published: Publishing message 8  
Published: Publishing message 9  
Published: Publishing message 10  
→ app-mqtt export $(egrep -v '^#' .env | xargs) && go run cmd/publisher/main.go  
Published: Publishing message 1  
Published: Publishing message 2  
Published: Publishing message 3  
Published: Publishing message 4  
Published: Publishing message 5  
Published: Publishing message 6  
Published: Publishing message 7  
Published: Publishing message 8  
Published: Publishing message 9  
Published: Publishing message 10  
→ app-mqtt export $(egrep -v '^#' .env | xargs) && go run cmd/publisher/main.go  
Published: Publishing message 1  
Published: Publishing message 2  
Published: Publishing message 3  
Published: Publishing message 4  
Published: Publishing message 5  
Published: Publishing message 6  
Published: Publishing message 7  
Published: Publishing message 8  
Published: Publishing message 9  
Published: Publishing message 10  
→ app-mqtt export $(egrep -v '^#' .env | xargs) && go run cmd/subscriber/main.go  
# gophercon/app-mqtt/mqtt  
mqtt/client.go:245:4: c.FixedHeader undefined (type *Client has no field or method FixedHeader)  
mqtt/client.go:245:26: c.Type undefined (type *Client has no field or method Type)  
mqtt/client.go:245:36: undefined: f  
→ app-mqtt export $(egrep -v '^#' .env | xargs) && go run cmd/subscriber/main.go  
Received message: Publishing message 1  
Received message: Publishing message 2  
Received message: Publishing message 3  
Received message: Publishing message 4  
Received message: Publishing message 5  
Received message: Publishing message 6  
Received message: Publishing message 7  
Received message: Publishing message 8  
Received message: Publishing message 9  
Received message: Publishing message 10  
Error reading fixed header: read tcp 10.62.105.23:61009->91.121.93.94:1883: i/o timeout  
Failed to receive PINGRESP: read tcp 10.62.105.23:61009->91.121.93.94:1883: i/o timeout  
Connection lost. Attempting to reconnect...  
Reconnected successfully  
Connection lost. Attempting to reconnect...  
Reconnected successfully  
Failed to resubscribe to /tsavo/west/mammal/gopher_2024: failed to read SUBACK: read tcp 10.62.105.23  
:61022->91.121.93.94:1883: i/o timeout  
Error reading fixed header: read tcp 10.62.105.23:61022->91.121.93.94:1883: i/o timeout  
Error reading fixed header: read tcp 10.62.105.23:61022->91.121.93.94:1883: i/o timeout  
Error reading fixed header: read tcp 10.62.105.23:61022->91.121.93.94:1883: i/o timeout  
Failed to receive PINGRESP: read tcp 10.62.105.23:61022->91.121.93.94:1883: i/o timeout  
Connection lost. Attempting to reconnect...  
Connection lost. Attempting to reconnect...  
Connection lost. Attempting to reconnect...  
Reconnected successfully  
Connection lost. Attempting to reconnect...  
Reconnected successfully  
Reconnected successfully
```

What is working ATM

- Animal tagging devices programmed with TinyGo and is being developed on Arduino nano 33 IoT
- Animal Location tracking and health monitor stats shared with MQTT
- The trial phase was completed with much success.

"Last Will"

Thank you, asante ...