

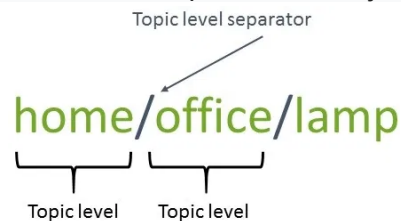
Aplicație pentru monitorizarea resurselor SO (Client MQTT)

-Documentație-

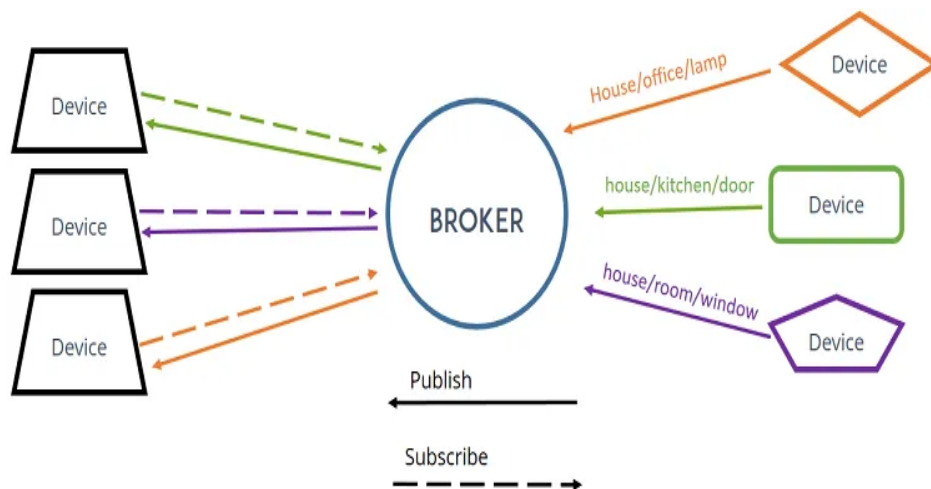
- MQTT este un sistem ușor de publicare și abonare în care se pot publica și primi mesaje în calitate de client.



- **Mesajele** sunt informațiile care circulă între dispozitive, fie ele comenzi sau date.
- **Topicurile** reprezintă modul de înregistrare al interesului pentru mesajele primite sau modul de specificare a locației în care va fi publicat mesajul.



- **Brokerul** este responsabil pentru primirea tuturor mesajelor, filtrarea mesajelor și decide cine este interesat de acestea ca apoi să publice mesajul către toți clienții abonați.



- Pentru implementare: **Mosquitto broker on the Raspberry Pi**

1. Două view-uri

- unul pentru publicare
- unul pentru abonare

2. Autentificare cu utilizator și parolă

- Autentificarea cu nume de utilizator și parolă este obișnuită pe toate sistemele computerizate, iar brokerul Mosquitto MQTT acceptă acest mecanism de autentificare.
- Pentru a utiliza autentificarea prin parolă, este necesară configurarea brokerului MQTT pentru a solicita acest lucru.

3. Listă de abonare configurabilă (creare/stergere din GUI, fisier config.)

4. Publicare manuală (din GUI) sau automată (periodic, configurabil) pentru valorile monitorizate

5. Implementare mecanism KeepAlive

- Mecanismul KeepAlive este utilizat pentru verificarea conexiunii dintre client și broker. Acesta se asigură că conexiunea este încă deschisă, dar și de faptul că atât clientul, cât și brokerul știu că sunt conectați.
- Atunci când clientul stabilește o conexiune cu brokerul, clientul comunică un interval de timp în secunde (60s by default) către broker. Acest interval reprezintă perioada maximă de timp în care cei doi ar putea să nu mai comunice.
- Dacă această perioadă este depășită de către client atunci acesta trebuie să trimită un pachet **PINGREQ** la care brokerul va răspunde cu un pachet **PINGRESP**. Dacă brokerul nu primește acest mesaj în cel mult $1.5 * \text{KeepAlive}$ interval atunci se va deconecta. În același mod, clientul se poate deconecta dacă nu primește mesajul de la broker.
- În cazul în care KeepAlive este 0, sistemul KeepAlive este dezactivat.
- Se folosește **connect** pentru a transmite informația despre KeepAlive brokerului

6. Implementare QoS 0,1,2

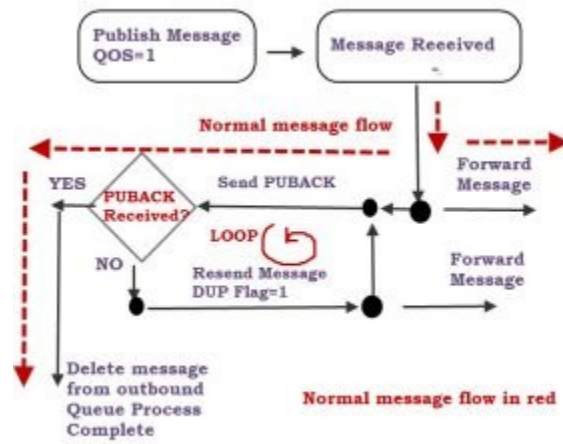
- QoS (Quality of Service) se referă la conexiunea dintre client și broker și sunt o modalitate de garantare a livrării mesajului. Există 3 nivele: QoS 0 (once), QoS 1 (at least once), QoS 2 (only once).

QoS 0:

- cea mai rapidă și nesigură metodă
- mesajul nu este stocat de către client deci este șters odată cu trimiterea din outbound queue
- nu se pot duplica mesajele

QoS 1:

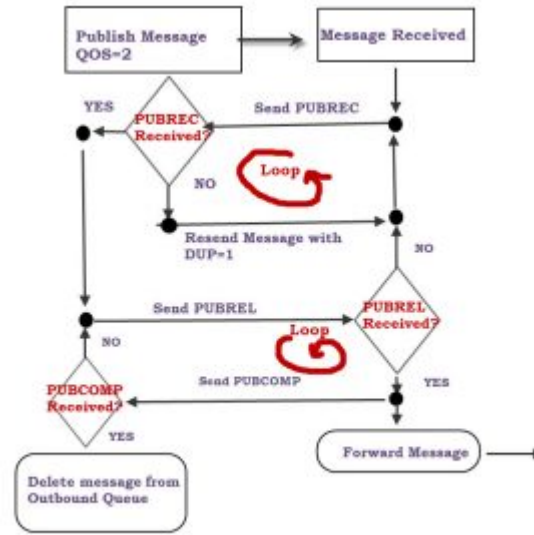
- mesajele pot fi livrate de mai multe ori
- după ce este trimis un mesaj se așteaptă confirmarea(**PUBACK**). după ce o primește șterge mesajul din outbound queue
- în caz contrar, retrimite mesajul cu **DUP flag set** la intervale regulate pana primește confirmare.



MQTT QOS 1 Message Flow Diagram

QoS 2:

- garantează livrarea mesajului o singura data
- cea mai înceată metoda
- executa ca la QOS 1, dar după ce expeditorul primește confirmarea mai trimite un mesaj(**PUBREL**)
- La primirea acestui mesaj, receptorul poate trimite mai departe mesajele (**PUBCOMP**)
- În cazul în care receptorul nu îl primește, acesta trimite **PUBREC**
- la primirea **PUBCOMP** procesul este complet și mesajul se poate șterge din outbound queue



MQTT QOS 2 Message Flow Diagram

7. Implementare mecanism Last Will

- **LWT- Last Will and Testament** este o modalitate excelentă de a notifica alți clienți abonați despre pierderea neașteptată a conexiunii unui alt client sau pentru a notifica abonații în legătură cu oprirea neașteptată a publisherului (exp. network error)
- Fiecare client își poate specifica ultimul mesaj de testament atunci când se conectează la un broker. Ultimul mesaj de testare este un mesaj MQTT normal cu un topic, **retained message flag**, QoS și payload.
- Brokerul stochează mesajul până când detectează că clientul s-a deconectat. Ca răspuns la deconectare, brokerul trimite ultimul mesaj către toți clienții abonați din topicul ultimului mesaj. Dacă clientul se deconectează grațios cu un mesaj corect DISCONNECT, brokerul renunță la mesajul LWT stocat.
- Publisherul îi transmite brokerului să notifice toți abonații unui topic folosind **last will message** în cazul în care conexiunea se oprește
- În cazul unei deconectări intenționate: **MQTT DISCONNECT**