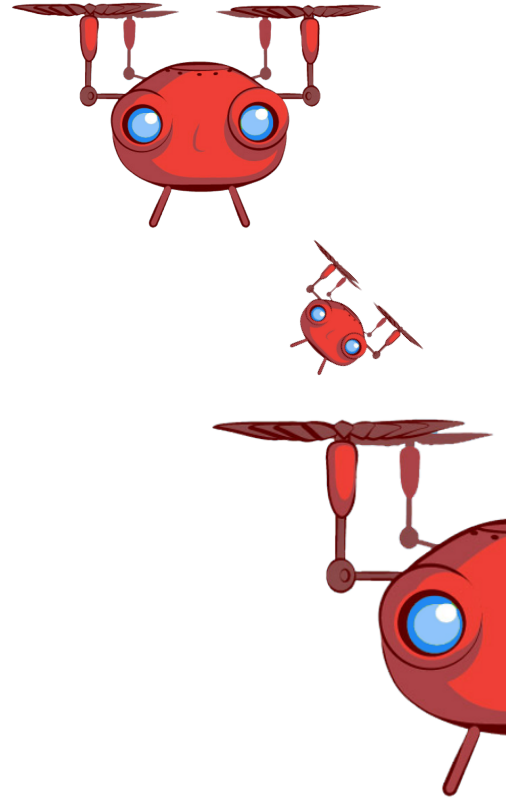
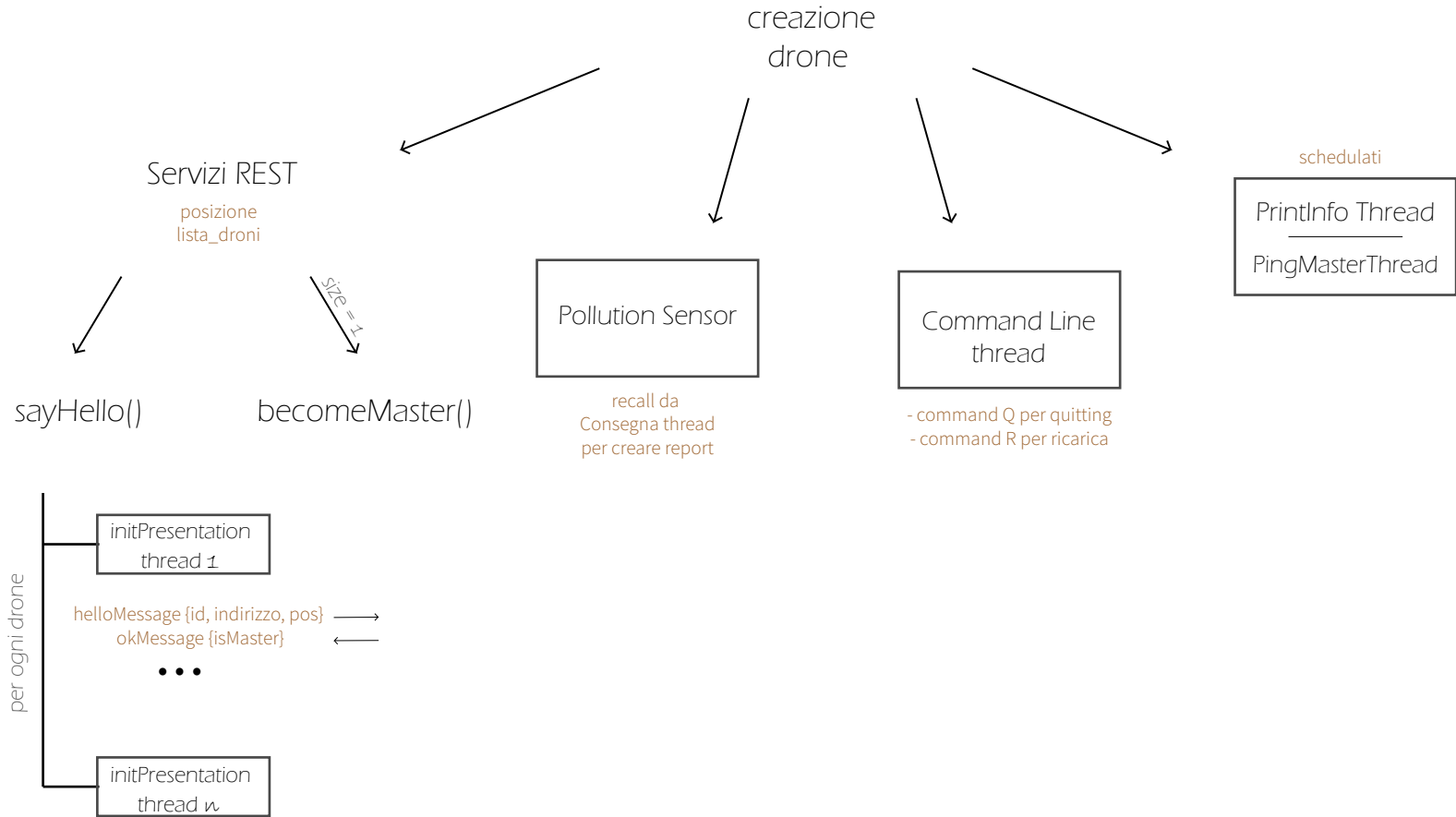


# DRONAZON

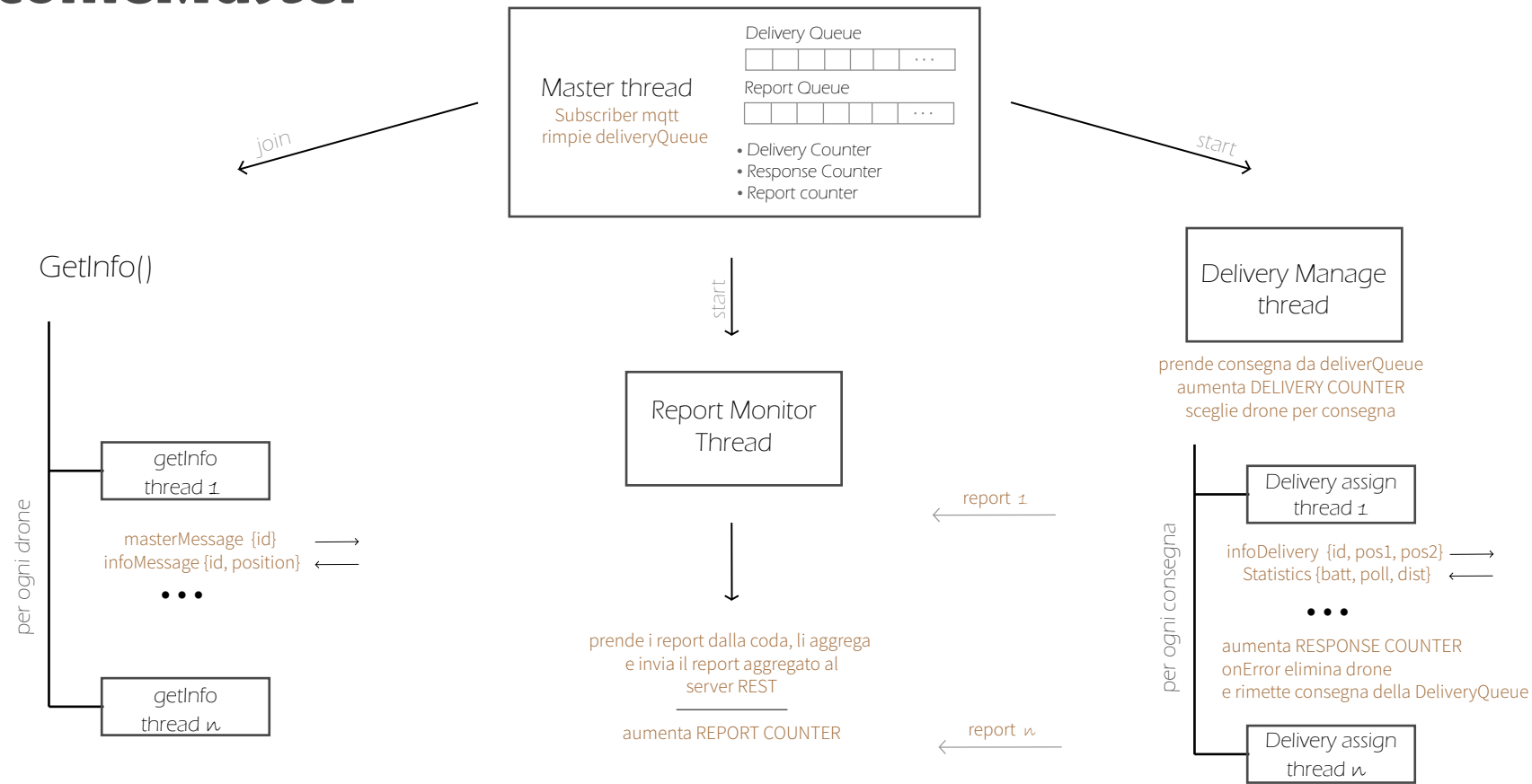


Naomi Demolli mat. 962322

# Inizializzazione



# becomeMaster



# Consegna

Master sceglie il drone migliore  
per la consegna e fa partire

start  
↓

Delivery assign  
thread

il master setta il drone non available

se il Master cade, inizia elezione

se il DeliveryDrone cade, viene rimosso  
dall lista e la consegna è rimessa nella  
DeliveryQueue

infoDelivery {id, from, to}  
statistics {time, batt, dist, poll}



Delivery Drone  
riceve l'ordine

Servizio consegna  
GRPC

drone si setta  
non available

join  
↓

Se la battery < 15 inizia  
la procedura di quitting

Aggiorna le statistiche del drone  
batteria, posizione, tot consegne

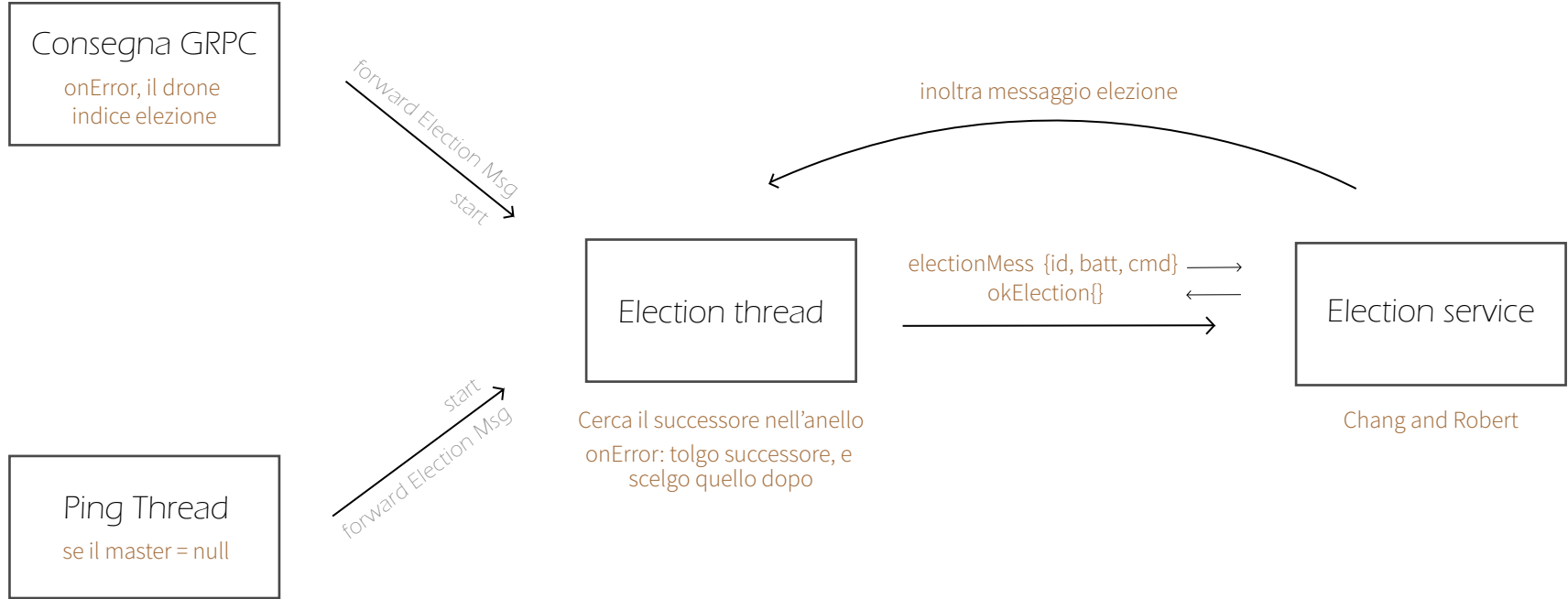
Drone settato available

↑

Consegna  
thread

Calcola le statistiche

# Elezione



# Elezione - nuovo drone durante election

## PROBLEMA A

Quando il drone 2, miglior candidato a diventare master, entra nell'anello non trova nessun master perché il drone 1 non ha ancora ricevuto il messaggio di "election 1", il drone 2 quindi indice l'elezione. Nel frattempo il drone 1 riceve il messaggio e diventa master, il messaggio "election 2" gira per l'intero anello, essendo 2 il candidato migliore, torna al drone B → ho due master!

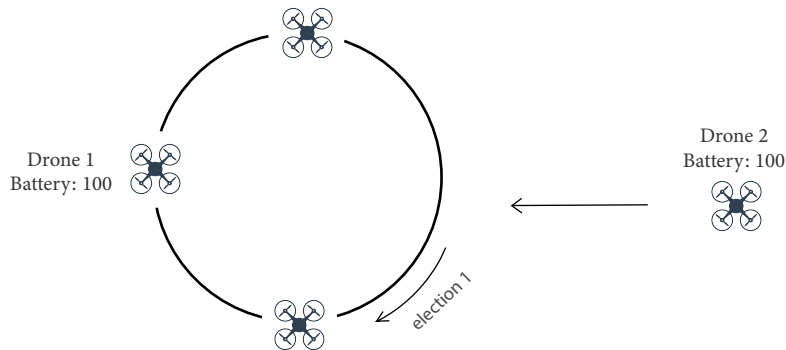
## SOLUZIONE

Quando un drone entra nella rete non indice subito un'elezione anche se non sente il master. Un nuovo drone si può accorgere dell'assenza del master in due modi:

- nel momento in cui si presenta alla rete, ma in questo caso non fa niente, aspetta che l'elezione sia indetta dai droni già all'interno
- il PingMasterThread, questo thread parte in ritardo all'inizio, dando il tempo al nuovo drone di settare il master

Nel caso non venga settato il master, il PingMasterThread si renderà conto prima o poi che il master non c'è e manderà un messaggio di "election 2":

- se il messaggio "election 1" è già passato, il drone 1 diventa master. Un drone già master che riceve un messaggio "election 2" scarta il messaggio e inoltra nell'anello un messaggio "elected 1" per sicurezza
- se il messaggio "election 1" non è già passato, il drone 2 si sostituisce nel messaggio e lo inoltra



# Elezione - nuovo drone durante elected

## PROBLEMA B

Il drone 2 entra nell'anello ma il messaggio "elected 1" è già passato. Il nuovo drone non saprà chi è il master eletto.

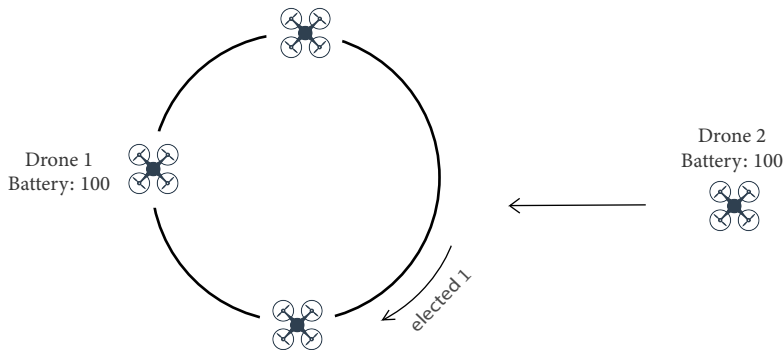
## SOLUZIONE

Un drone quando diventa master manda un messaggio - contenente l'ID del master - ad ogni drone, nella fase di presentazione.

Un drone che riceve questo messaggio e ha settato master = null, setta il master e manda la sua posizione.

Un drone diventa master dopo aver ricevuto un "election" con il suo ID, un nuovo drone che entra dopo questo momento capisce chi è il master nel momento in cui, entrando, si presenta agli altri droni della rete.

Nel caso - poco probabile - in cui il drone non sappia chi sia il master dopo la fase di presentazione di esso, il master prima o poi affiderà una consegna al drone 2, un drone che riceve una consegna da effettuare ma ha master = null, setta come master il drone che ha inviato la consegna e la porta a termine.

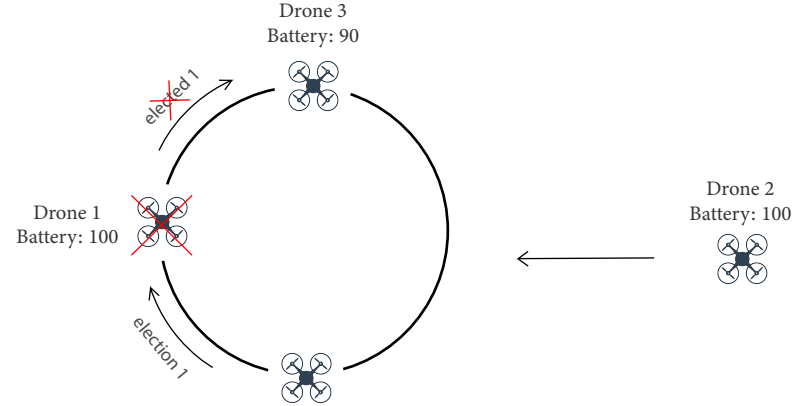


# Elezione - caduta master pre elected

## PROBLEMA C

Il drone 1 riceve il messaggio di "election 1" ma cade prima di poter inviare il messaggio "elected 1".

I droni nella rete sono settati participant = true ma nessuno scoprirà mai che il drone 1 è master → la rete è in stallo



## SOLUZIONE 1

Ciascun drone ha un PingMasterThread che controlla se il drone conosce o meno il master.

**Nella soluzione 1 il Ping Thread non controlla se il drone è già participant, se non trova il master indice un'elezione.**

Il drone 3 è il miglior candidato dopo il drone 1, il pingMasterThread del drone si accorge che il master non c'è, manda un messaggio di "election 3", il suo messaggio viene inoltrato lungo tutto l'anello - è il candidato migliore - fino a che il drone 3 diventa master.

Gli altri droni che si accorgono che non c'è un master mandano un messaggio di "election" ma ci sarà un altro drone nell'anello che è un candidato migliore con participant = true e quindi scarcerà il messaggio.

CONTRO: vengono indette più elezioni del necessario



# Elezione - caduta master pre elected

## PROBLEMA C

Il drone 1 riceve il messaggio di “election 1” ma cade prima di poter inviare il messaggio “elected 1”.

I droni nella rete sono settati participant = true ma nessuno scoprirà mai che il drone 1 è master → la rete è in stallo

## SOLUZIONE 2

Ciascun drone ha un PingMasterThread che controlla se il drone conosce o meno il master.

Nella soluzione 2 il Ping Thread controlla se il drone è già participant, se non è participant indice elezione.

```
while(drone.isParticipant()){  
    try {waitTime();}  
    catch (InterruptedException e) {e.printStackTrace();}  
}
```

PRO: vengono indette meno elezioni in condizioni normali

CONTRO: il problema C non viene risolto perchè nessuno inizierà una nuova elezione essendo tutti participant = true

Bisogna aspettare l'entrata di un nuovo drone e far valere un'assunzione: gli ID dei droni devono essere crescenti

**ASSUNZIONE:** ID sono crescenti, questo garantisce che il nuovo drone sarà sempre il candidato migliore e il suo “election” girerà per l'anello

# Elezione - caduta master in election

## PROBLEMA D

Il drone 1, il miglior candidato, manda il messaggio "election 1" ma cade.

Il messaggio verrà inoltrato da ogni nodo dell'anello senza fine.

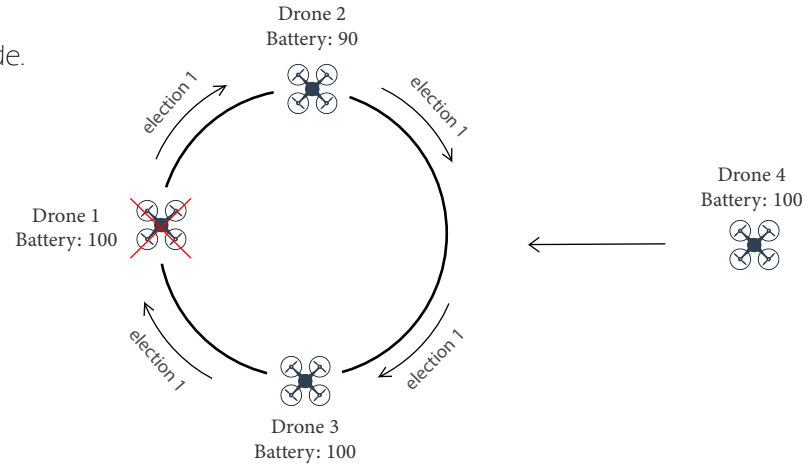
## SOLUZIONE 1

Se il Ping Thread **non controlla participant**, prima o poi partirà un'altra elezione dal miglior drone rimasto, portando all'elezione: un drone che diventa master e riceve il messaggio di election 1 lo scarta.

## SOLUZIONE 2

Se il Ping Thread **controlla participant** - se sono tutti participant - nessun drone nella rete farà partire un'elezione e il messaggio continuerà a girare e nessun drone verrà eletto.

Bisogna attendere che un nuovo drone entri nella rete e che valga l'assunzione sugli ID: il nuovo drone indice un'elezione, essendo il candidato migliore, il suo messaggio viene inoltrato da tutti, diventerà master e alla ricezione di election 1 lo scarcerà.



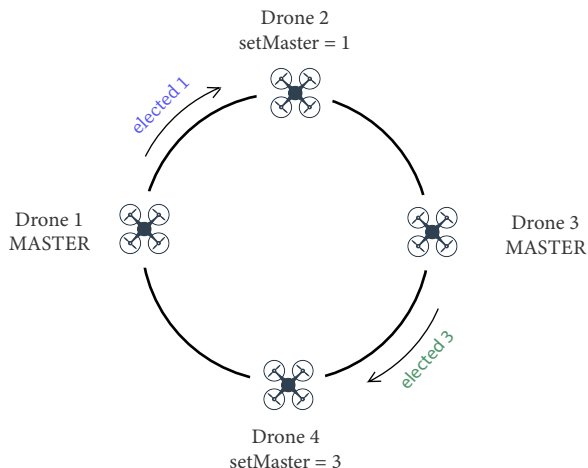
# Elezione - strategia di recovery

## PROBLEMA A

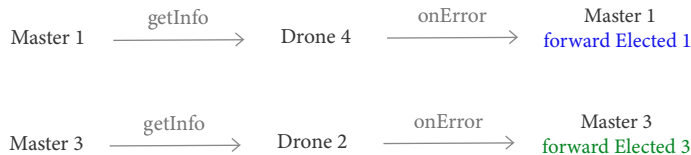
Il master appena eletto si presenta ai droni della rete con il proprio ID per avere le posizioni e la batteria aggiornate, ogni deliveryDrone che riceve questo tipo di richiesta esegue un controllo tra l'ID del master che ha salvato a fine elezione e l'ID del master che si sta presentando. Se gli ID non coincidono l'elezione si è conclusa erroneamente.

## RECOVERY

Il drone che rileva l'errore solleva l'onError del master con un'eccezione ErrorMasterID, il master che riceve un onError sull'ID del master inoltra un messaggio di "elected" con sè stesso.



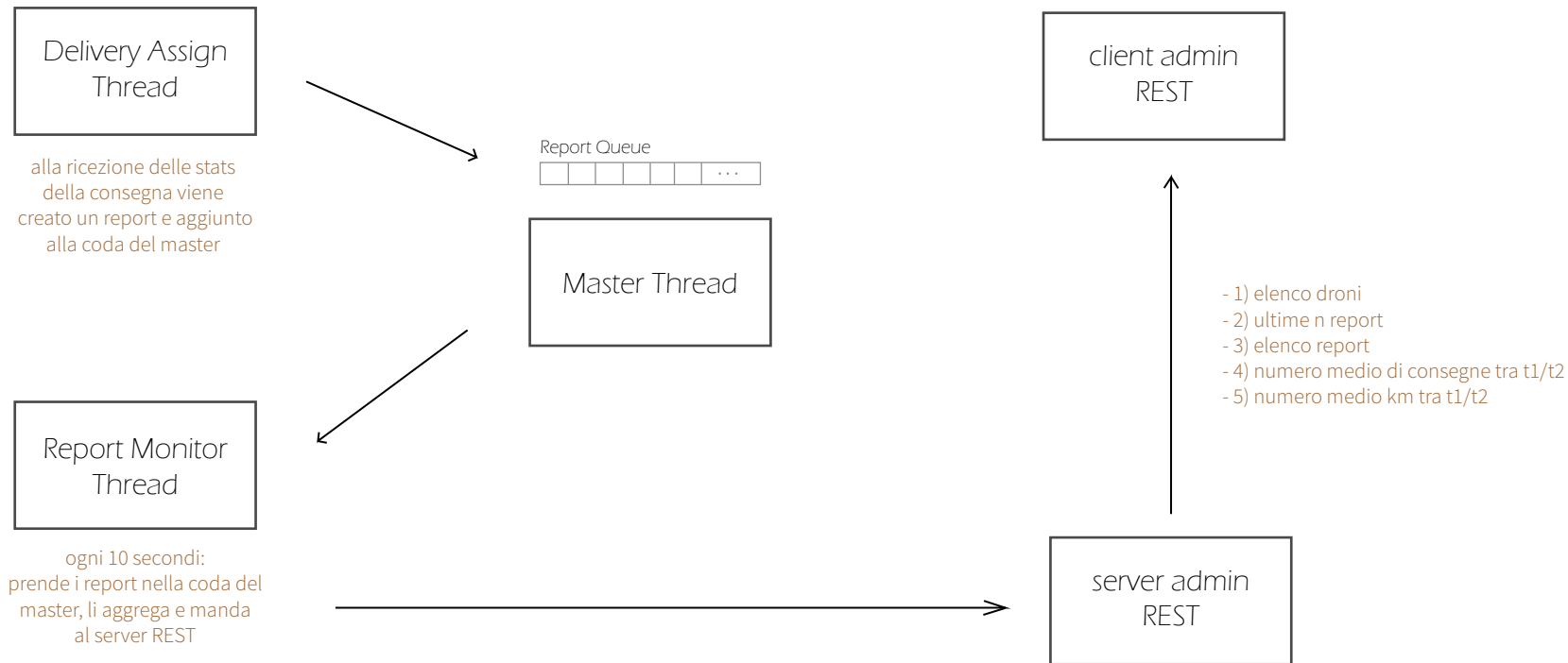
## Esempio: due droni diventano master



Master 1 riceve elected 3 e capisce che c'è un altro master e che è migliore rispetto a lui si setta master = false e Quitta i servizi da master (sub Mqtt, Delivery Manage, report Monitor) e inoltra il messaggio elected 3

Master 3 riceve elected 1 ma vede di essere il master migliore tra i due, sostituisce il suo ID nel messaggio di elected e lo inoltra al drone successivo

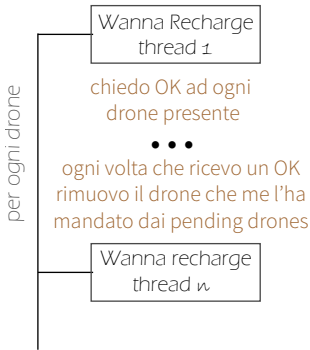
# Report



# Ricarica

Command Line Thread

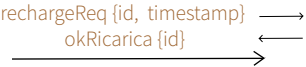
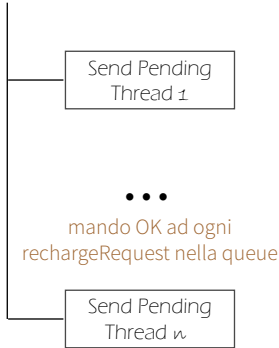
Inserire 'R' e viene  
settato wannaCharge



Recharge Monitor Thread

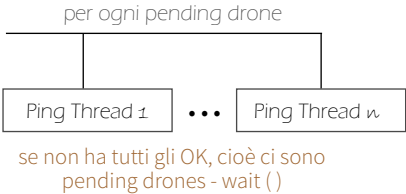
Recharge Request Queue

						...
--	--	--	--	--	--	-----



Recharge Service

Il drone riceve una richiesta di recharge:  
- se è in elezione, aspetta a rispondere  
- Ricart and Agrawala



Il drone controlla periodicamente se  
i pending drone sono caduti,  
se un drone è caduto non  
aspetto il suo OK

se il drone ha tutti gli OK,  
se non sta consegnando

Recharge

il drone si aggiorna  
posizione = [0, 0]  
battery = 100

End Recharge

notifyMaster ( )

Il master setta nella sua lista:  
- inCharge = false  
- Available = true;  
- Position = [0, 0]  
- Battery = 100

# Ricarica - Elezione - Consegna

## DRONE IN ELEZIONE

Il drone riceve una richiesta di recharge	Risponde quando finisce elezione
Il drone vuole ricaricarsi	Ricarica quando finisce elezione
Al drone è affidata una consegna	Se in elezione, non c'è master

cerco di posticipare la ricarica  
a dopo l'elezione

## DRONE IN RICARICA

Drone riceve messaggio election	Drone partecipa con battery = 100
Al drone è affidata una consegna	Notifica il master di essere inCharge

getUpdateBattery ( )

## DRONE IN CONSEGNA

Drone vuole ricaricarsi	Ricarica quando finisce la consegna
Drone riceve messaggio election	Partecipa all'elezione con battery - 10

getUpdateBattery ( )

# Lock

## • Available Lock



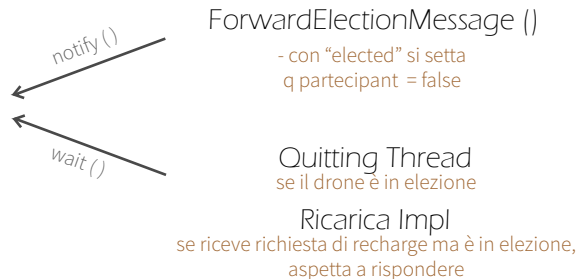
## • Choosing Drone Lock



## • Response counter Lock



## • Participant Lock



## • Report counter Lock



## • Update Drone Lock

