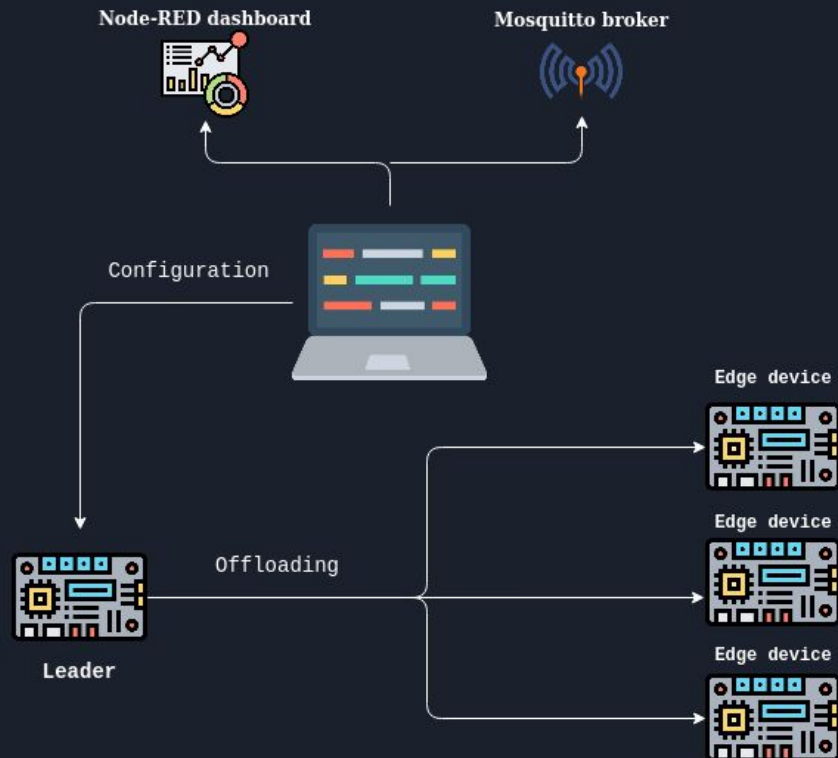
The background is a dark blue gradient. On the left, there is a large, semi-transparent circular image of a circuit board. Overlaid on the top left of this circle are two overlapping triangles: a blue one in front of a green one. In the top right corner, there is a 3D perspective view of a circuit board's surface, showing numerous small components and traces.

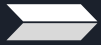
Framework to load
balance data analysis
on the device or at
the edge

System diagram





Tools



Mosquitto (mqtt)



Node-Red (dashboard)



VSCode with PlatformIO (embedded development)



paho mqtt (python library)

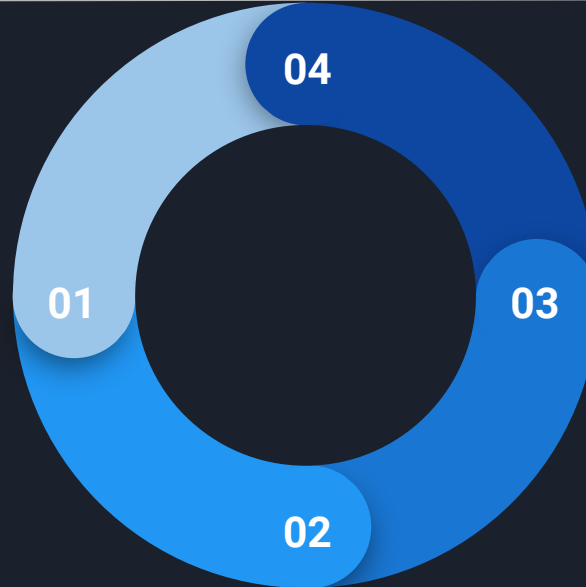
Flowchart

Configure the problem

Choose the type of problem, its input and the number of iterations per second

Load balancing

Determines if and how to assign load to edge devices



Show results

View results and performance on the dashboard

Run the problem

Each device performs the required problem



System's features

- 01 Dynamic Wifi credentials
- 02 OTA Updates
- 03 Ability to detect the computational capabilities of any device
- 04 Distribute computation to edge devices if the computational capabilities of the leader aren't enough



System's features

- 05 It works fine even if an edge device runs out of battery or disconnects from the network
- 06 Monitor the status of the mqtt broker constantly
- 07 Dashboard to monitor workload distribution
- 08 Stop the system in case of too high latency



Persona 01

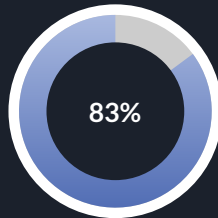
Load distribution

As long as the leader is able to solve the number of iterations of the required problem, then no other device is involved.

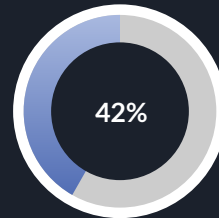
Otherwise, the leader carries out all the iterations he can and distributes them to the edge devices based on their computational capabilities: the more powerful a device is, the higher the number of iterations assigned to it will be



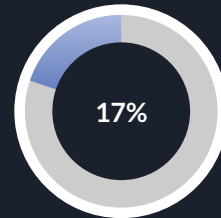
Leader



Edge1



Edge2



Edge3



Thanks for the attention