Workflow pseudocode of the load balancer.

```
Algorithm LoadBalancer(r):
if r.latency == ‘high’ then       // latency-sensitive
    if onboard.resource_usage < 80 then
        Process(r, "onboard")
    else
        if bandwidth_to_cloud == 0 then
            Process(r, "onboard")
        else if 0 < bandwidth_to_cloud <= 5 then
            if r.priority == "accuracy" then
                Process(r, "onboard")
            else
                r.resolution = lower(r.resolution)
                Process(r, "cloud_low_res")
            end if
        else if bandwidth_to_cloud > 5 then
            Process(r, "cloud")
        end if
    end if
else
    // Non-latency-sensitive request
    if bandwidth_to_cloud > 5 then
        Process(r, "cloud")
    else
        Queue(r, "onboard")
    end if
end if
```

Thresholds Values

Decision Type	Key Thresholds	Action
Onboard preferred	CPU < 80%, RAM < 80%, GPU < 80%	Process onboard
Bandwidth low	Bandwidth < 5 Mbps	Lower resolution or onboard
Bandwidth good	Bandwidth ≥ 5 Mbps	Offload to edge/cloud
Edge/cloud allowed (extra)	Download speed > 5 Mbps	Safe to offload

List of applications with varying requirements for load balancer.

Category	Application	Reason
Onboard	Traffic light, lane detection	Ultra-low latency, critical safety
Offload	Traffic-sign detection	High compute, medium latency tolerant
Lower resolution/priority	Collision detection, depth estimation	can run at lower resolution