Idea

- An edge data center that is also an orchestrator should be able to elect a leader among the other edge data centers.
- A leader (passively known to cover such a role by not having a leader) can receive tasks and offload them to its subjecteds.
- If there is no computational power left the task is offloaded on the Cloud.

Implementation

- I started by creating a custom class that extends DefaultDataCenter using the proposed ClusterEdgeDevice as a base.
- In startInternal a task with tag LEADER_ELECTION is scheduled with INITIALIZATION_TIME + 1 delay.
- In process event the custom tag is caught.
- There is a check for:
  - the device is an edge data center,
  - the device is an orchestrator,
  - "LEADER" is the orchestration method,
- If everything is true then the method **leader** is called.
  - in a loop every datacenter is taken into account and is checked whether it:
    - is not the same data center as the one evaluating,
    - is an edge data center
    - the distance between the two data centers is smaller than the range of the edge data centers.
  - If it is the case then there is an evaluation regarding the MIPS (I used that criterion for electing a leader).
    - If the MIPS of the candidate are greater than the evaluator
    - and the max of the data center seen until that point is lower than the MIPS of the candidate

– then the candidate becomes the (potentially temporary) leader, the max is set to leader's MIPS.
– Upon ending the loop if a leader has been found the current datacenter is added to its subjected list. This will be used when a new task must be executed by someone with enough free computational power.