Abstraction for IoT Scenarios

Learning from couple of scenarios for IoT, we tried to come up with an abstraction of the system which is not tied to a specific use case, instead represents a generic use case and can be overridden to meet the needs of a specific use case. The system is governed by these 3 variables:

1. State : state could be any choice of single/multiple values to represent the current situation of the system
2. Event : Any change in one or more values of a state variable is an event
3. Action : Action are changes in the system taken place after occurence of every event, or events act as a trigger for every device

Assumptions:

1. We classify the system to contain 2 types of objects:
    a. Devices:
        i. The IoT devices in the scenario.
        ii. Devices will report state changes to the manager.
        iii. Devices will receive actions from manager.
    b. Device Manager:
        i. Manager maintains metadata for all the devices
        ii. Manager computes the incoming state change information and decides which device to instruct to based on the user defined rules.
2. All the communication takes place from device communicating state change to device manager and device manager instructing for corresponding action
3. All the devices while implementing have a database layer and an API layer which would help in taking actions and communicating to the manager
4. The manager is considered to be assigned a static IP in the network so that a new/existing device can reach to manager using that IP.

Abstract Methods:

<u>Device:</u>

1. Report state change to Manager
2. Change in state variable
3. Register API on manager
4. Receive Actions from Manager
5. De-Register Device
6. Keep Alive
7. Update Metadata Device
8. Manager Metadata Update
9. State variable for system (continuous/discrete)
10. Mode x of operation(Burst/delayed)

Manager:

1. Register a device
2. Send Action to Device
3. Receive state change
4. De-Register device (Soft delete/ mark offline)
5. Heartbeat from device
6. Metadata Update from Device

Device Metadata:

1. IP address
2. Status (Online/Offline)
3. Device Operation Constraint (Ex. Lights may have a specific timing they can accept actions. No need to send actions to light when it is day time!)

Manager Metadata:

1. IP address

Input of the rules:

1. One YAML file per scenario.
2. YAML file parsed by a Python script and creates binaries for that scenario.
3. Aiming for two scenarios (Parking Lot and Thermostat)