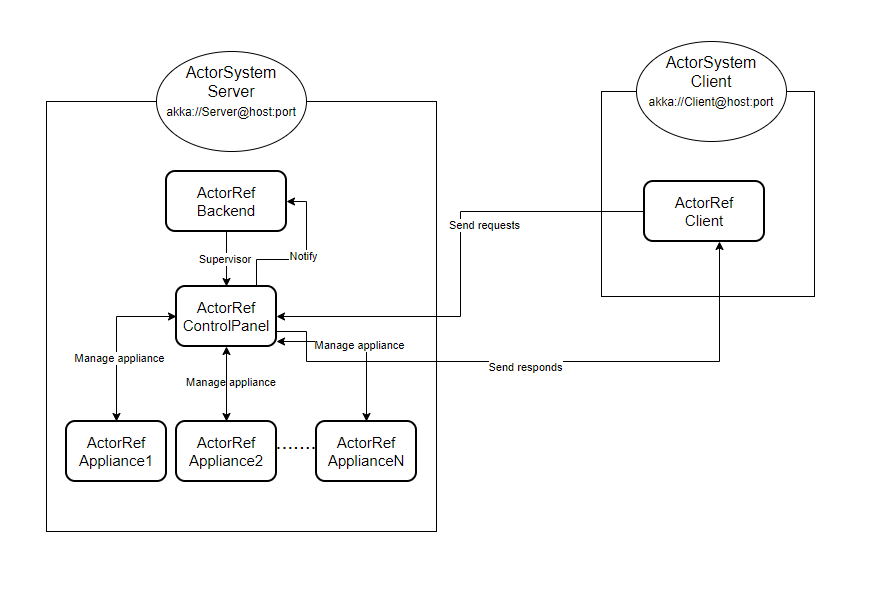
Project 3 Design Document

**Introduction**

The goal of the project was to create a software layer for a distributed smart homing system. This system must be able to manage appliances throughout the house, even from different rooms, allowing the user to enter preferences and achieve changes in the behavior of those appliances. In addition, the system must be covered against any process crashes.

The system uses akka's actor model and its key components are described below.

**Implementation**

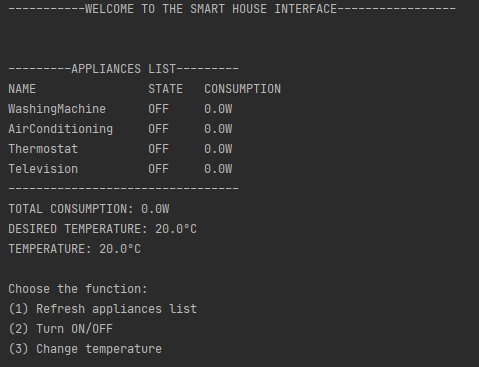


The diagram represents the basic structure of the system. It is defined by two main sections, the **Client** section, and the **Server** section.

**Client**

The term Client is defined as any interface that allows a user to interact with the system. In the diagram only one is represented but the number of clients connected to the system is not limited. Each of these interfaces represents an **ActorSystem** and is therefore identified by its own **address** and reference **port**.

First, the client gets the address of the control panel, which is the actor to which it sends requests, via its address. Once connected with the latter it receives the basic interface.



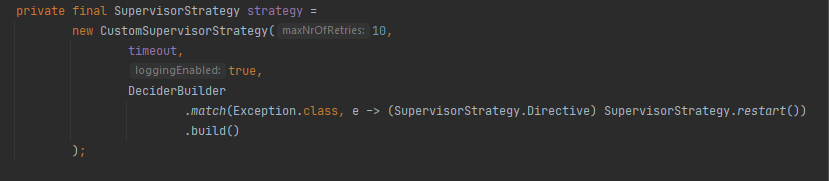
This is the interface that is presented to every connected client. Through it a user can:

* Observe and refresh the list of appliances connected to the system and their current status, i.e. whether they are on or off and the resulting power consumption.
* Turn on or off any of the equipment connected to the system.
* Change the desired temperature.

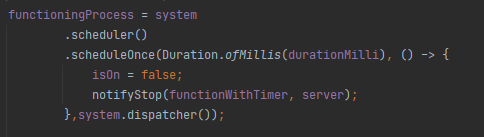
**Server**

The server site is composed by three Actors:

* The **Backend Actor** is the one responsible for overseeing the work of all the underlying actors. So its main purpose is to reset the actors it is supervising whenever one of them crashes or has any fatal error. It also logs the messages sent to it by the control panel.



* The **ControlPanel** actor is the one that connects with clients and receives requests from them. Each request corresponds to a function that the control panel performs and communicates to the level below it, that of the devices. As for the backend, it also supervises the actors below it.
* The **Appliance** actor is a generic term for any such actor that is connected to the control panel. They have very different functions but can be summarized in 3 different types:
  + Appliances that are turned on and off only by an explicit request from the user. They are the simplest type of appliance that only has a sensor that verifies the current consumption.
  + Appliances that have a timed operation. Once they are turned on, they stay on for a preset time and then turn off by themselves, notifying the control panel of the event. This is done using the akka scheduler class.



* + Devices that affect temperature. Temperature is a data that is collected by a special device, the **Thermostat**. This is the only one that has a sensor that detects the temperature of the house and through this device the control panel receives the temperature whenever it needs it.

In addition, this device, once turned on, takes care of heating the environment by increasing the temperature. The **Air conditioning** appliancedoes the opposite thing. Every time the user communicates that he wants to change the temperature, the control panel turns on the reference device in order to heat or cool the room until the desired temperature is reached. Once this is done it turns off automatically.

Also in this case the akka scheduler is used, but unlike the previous case, before shutting down, it performs the action of adjusting the temperature of one degree at a time until it reaches the desired one.

