# Problem Statement –

* A multi-threaded (e.g. file-based) web server with thread-pooling

implemented in Java.

* Extension: Add proper HTTP/1.1 keep-alive behavior to your implementation based on the http-client's capabilities exposed through its request headers.
* A JavaScript application simulating house automation: pressing a

button on a control panel would visually turn on a light, change the

temperature or close the curtains. Some constraints:

* + the application must use jQuery
  + the components must have HTTP based "server" interaction (use a static file
* for simplicity, data persistence is not required). For example, the heating component

retrieves the current temperature from the server and also sends the desired one

back to the server. the solution has to be extensible and documented, so that we can develop our own components that react to events

* The application will be executed on a plain HTTP server with no possibility to run

code server side and is being viewed in 2 major browsers of your choice.

# Approach-

Leveraging the Spring Boot Embed Tomcat Http Server.

1. Solution needs to be capable of supporting multiple UI Clients like Mobile,Desktop,IPAD etc..
2. We need platform independent Service Layer which can be consumed by Multiple Clients.
3. Spring comes with wide variety of integration options from back end. like Messaging, Mongo Db etc. with Spring Boot and MVC can create service Layer.
4. Spring Follows Jax-rs Rest standards.
5. Resource Centric and Can easily fit in to the frameworks.
6. Sling from AEM can also be leveraged as Web application framework along with Jetty Embed server . <http://sling.apache.org/documentation/the-sling-engine/the-sling-launchpad.html>
7. Horizontal Scaling(HA) possible with Spring&Tomcat implementation.

Spring Static directory can serve the static files like JS/CSS and html.

JQUERY used as a Javascript framework to make backend POST/PUT/GET calls.

JSON used as communication mechanism between Client and server.

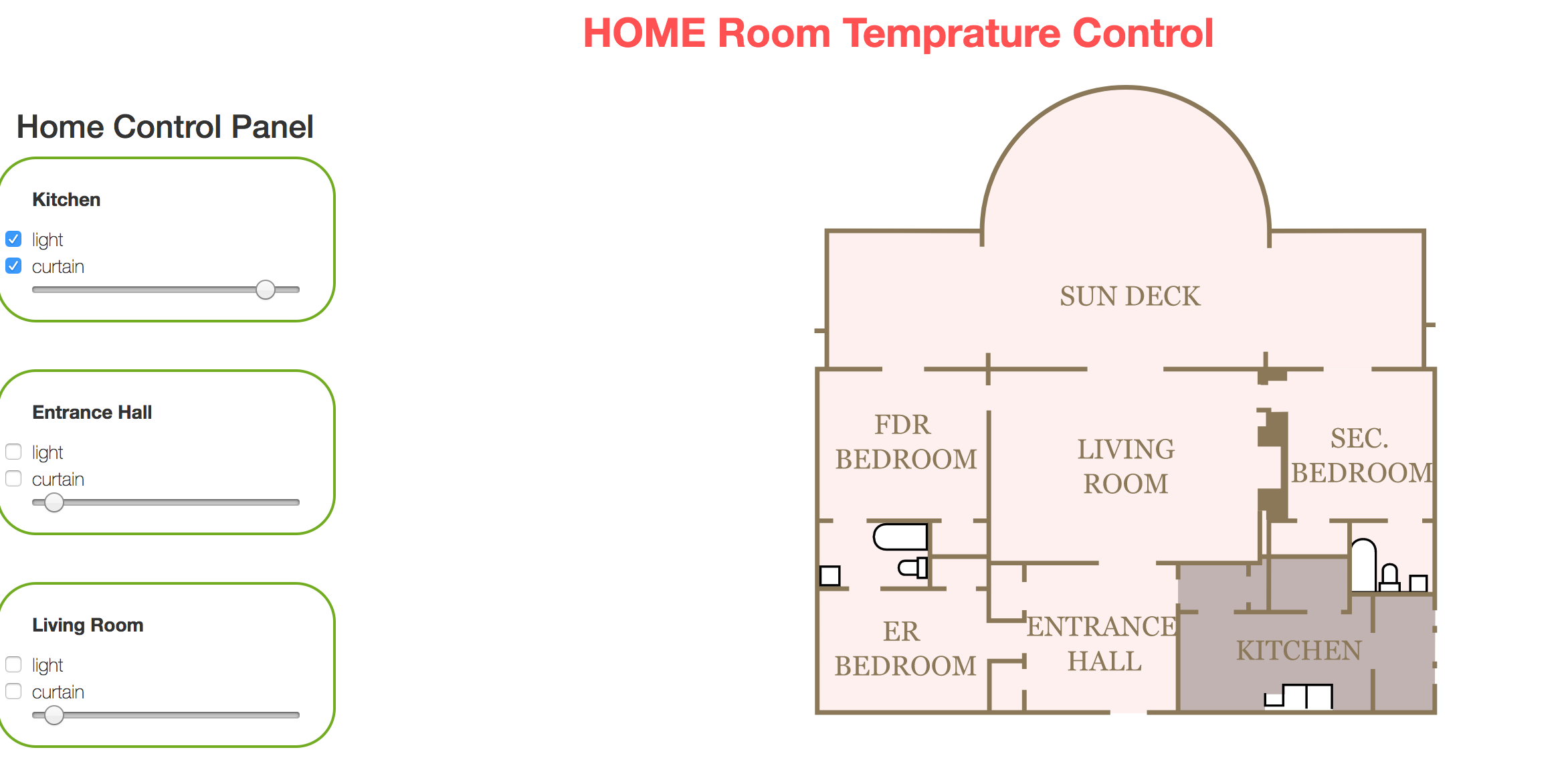
Home Control Dashboard

<http://localhost:8090/index.html>

Backend JSON response calls-

<http://localhost:8090/v1/rooms>

<http://localhost:8090/v1/rooms/path5621>



SVG file has room specific id’s . Can change the color of room with style.fill.

If there is any change in the Temperature or light values will be persisted in Backend.

Using HSL pattern changing the desired color.

Solution as extendible – As Backend we follow synchronous/asynchronous pattern to integrate with Hardware systems to control the light and temp automatically.

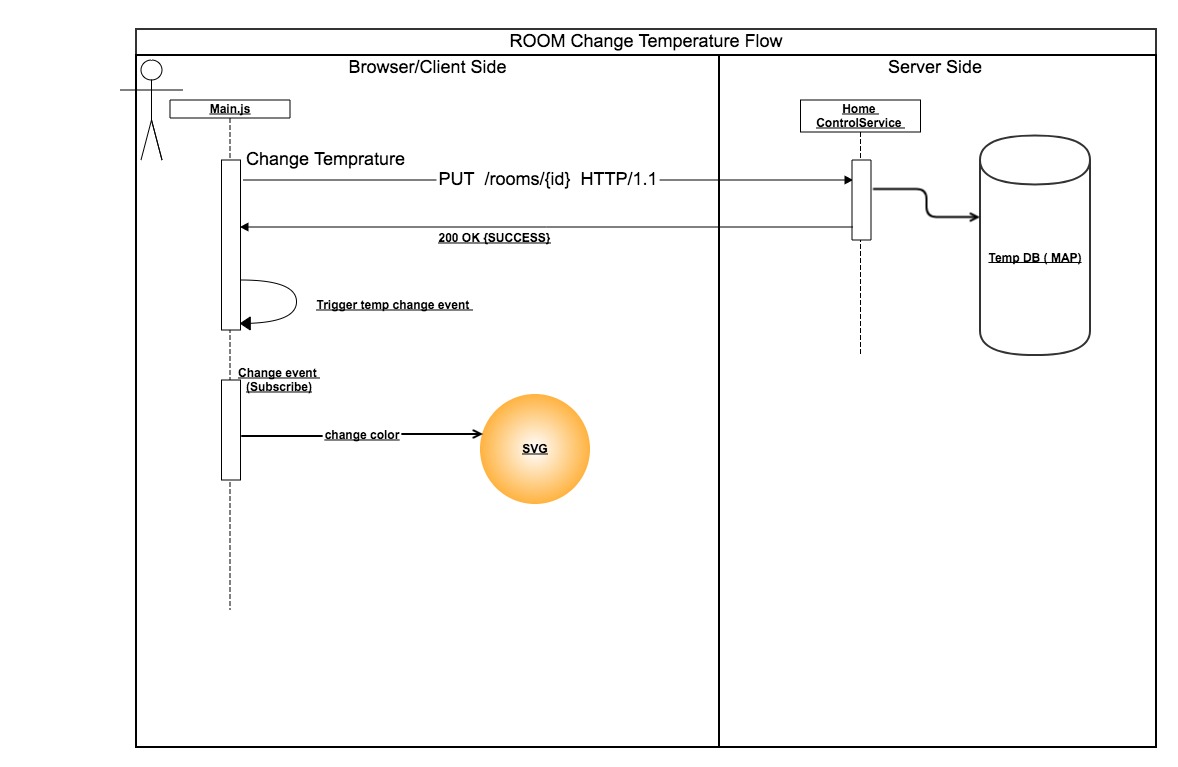
**public** RoomResonse updateRoom(String roomId ,RoomRequest roomRequest) **throws** APIException

It has mechanism to integrate and return status to front end.

From Front end point of view Listen to custom events to do necessary actions.

*$(document).on("HouseLightAndCurtainEvent", function (evt) { });  
$(document).on("TempChangeEvent", function (evt) { });*

Here is the high-level flow to change the Temperature –



Physical architecture to handle HA&DR patterns –

