

 **Edit**

Jeremy Peer Review

[Jeremy Clark](#)

5

Hi Guys - this is my first draft of a class diagram. I have a more fleshed out Design doc, but wanted to start with this diagram. I've added a bunch of notes containing summary information that will ultimately go in the class dictionary - but wanted to offer some insights into my design here. I've definitely abstracted certain pieces in a potentially strange way (with my 'interface-based' approach to describing devices), so definitely ask any questions or provide any feedback you see fit!

[House Mate Model Service.png](#)

Search entries or author

Unread

 **Subscribed** **Reply**<https://canvas.harvard.edu/groups/33458/users/4605>[Vinay Mishra \(https://canvas.harvard.edu/groups/33458/users/4605\)](https://canvas.harvard.edu/groups/33458/users/4605)

Sep 24, 2017

Hi jeremy,

I understand the intent of your design and it is somewhat similar to what I had in mind except interfaces for devices. I liked the idea of having occupants information with the service itself. I am still reviewing your design and if I find any requirement not captured I will let you know. One not so important thing: add auth_token in interfaces functions.

Edited by [Vinay Mishra \(https://canvas.harvard.edu/groups/33458/users/4605\)](https://canvas.harvard.edu/groups/33458/users/4605) on Sep 24 at 12:18pm

 **Reply**<https://canvas.harvard.edu/groups/33458/users/78924>

Sep 25, 2017

Thanks, Vinay. Helpful feedback. If you have any more thoughts pass them my way!

 **Reply**<https://canvas.harvard.edu/groups/33458/users/96884>[Michael Blanchard \(https://canvas.harvard.edu/groups/33458/users/96884\)](https://canvas.harvard.edu/groups/33458/users/96884)

Sep 24, 2017

Hey Jeremy,

Looks great! I think the one big thing that is missing are exceptions (and authtokens which was already mentioned). You'll have to figure out how to deal with bad commands. I'm guessing the fully qualified name will be a good tool to check for exceptions with. The other thing, and I haven't quite figured out the details yet myself, is that I'm struggling to see how various actions on devices get back to the knowledge graph in the form of predicates. I'm guessing that is where availableModes() is going to play in. availableModes() could be turned into predicates in the knowledge graph. That's at least what I'm planning on doing as I have a similar list. In createAppliance/Sensor() of the service class I would think you could send that list to the knowledge graph with the ID of the newly created device.

Also I think I like the way you made House a separate class from the HouseMateModelService. I did it differently thinking that a lot of those commands such as create house would be left to the other layers. I'm going to look at the design docs again. But it probably makes sense if the person has multiple houses to do this.

I'm going to keep looking it over though over tomorrow as I continue to think about these problems. More feedback to come!

Mike

[← Reply](#)

https://canvas.harvard.edu/groups/33458/discussion_topics/244654

Sep 25, 2017

Thanks, Mike - helpful feedback. Definitely need to work on exceptions (and authtoken) integrations. The questions you raise are good ones: I think the fqcn will be used to look up a object in the house graph, and check whether the 'status' parameter passed in the CLI is an available option...if it is, the predicates will always reflect the expression defined by the interface type. This could definitely be clearer in the document. But ultimately, in this design, all state will be stored in the KG, and the House graph will just be used to define the House layout and what CLI keywords each node responds to.

[← Reply](#)

https://canvas.harvard.edu/groups/33458/discussion_topics/244654

Sep 25, 2017

Ahh I see so like here in the CLI examples:

```
# show all state for the smoke detector
show sensor house1:kitchen1:smoke_detector1
```

That is basically your fqcn right there. Makes sense.

 Reply