FooSH

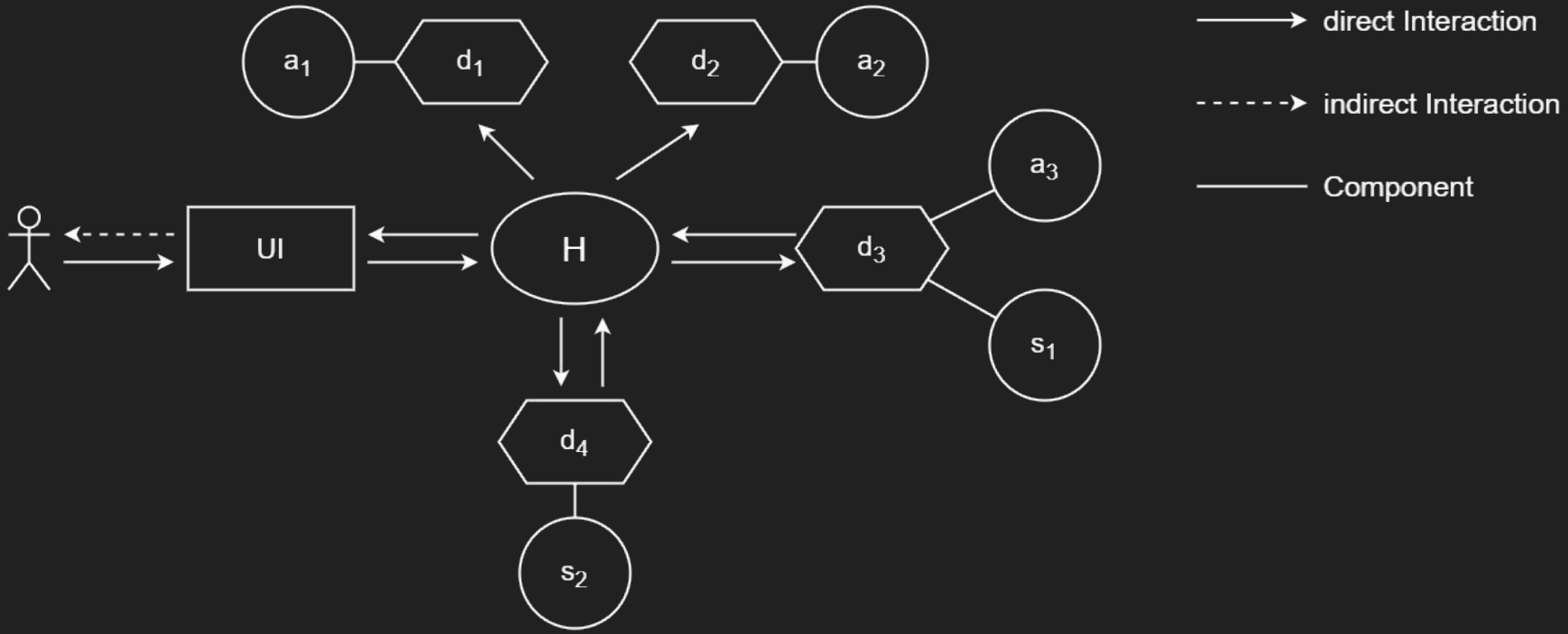
A Framework for outcome-oriented Smart Homes

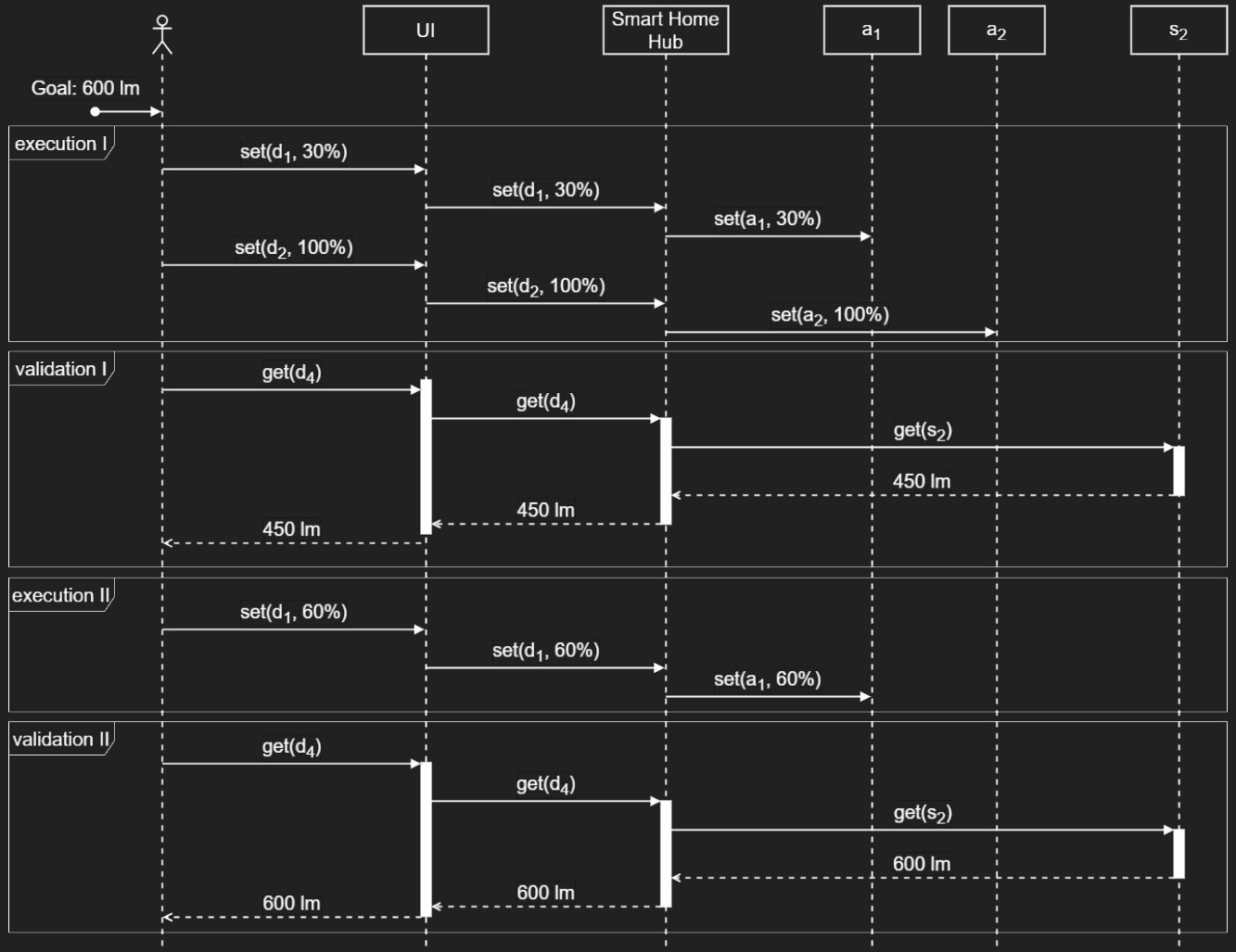
Master's Thesis Colloquium

Malte Josten
Applied Computer Science

What makes up a Smart Home?

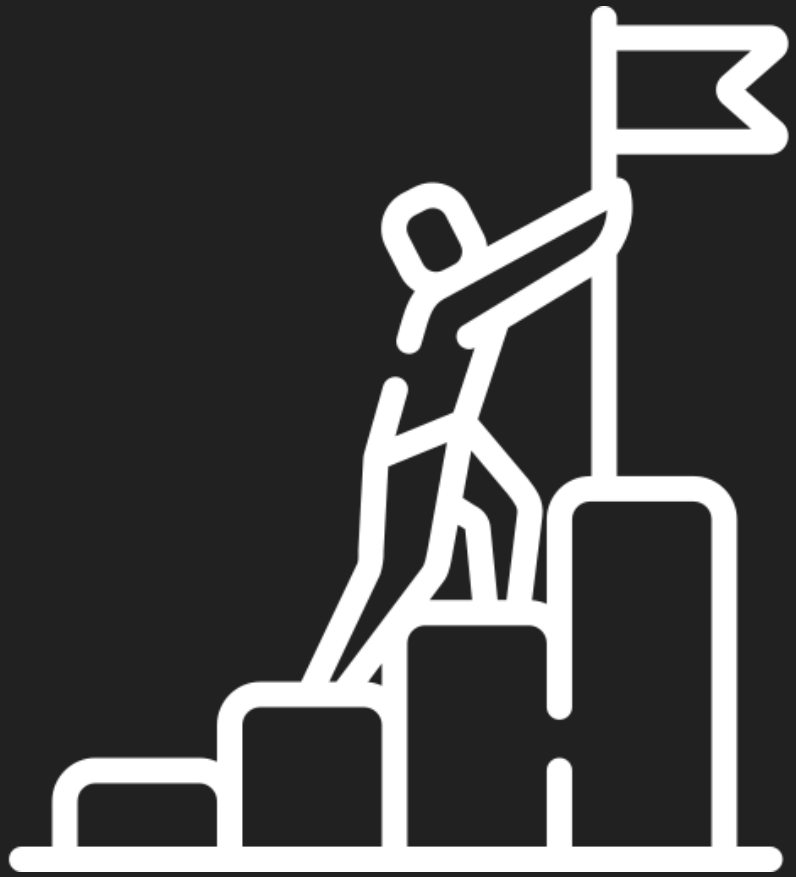
- A gateway or hub H
- A set of smart things $D \coloneqq \{ d_1,\dots,d_u \mid u \in \mathbb{N} \}$
  with each smart thing $d \coloneqq (\{ a \mid a \in A \}, \{ s \mid s \in S \}) \in D$
- A set of actuators $A \coloneqq \{ a_1,\dots,a_v \mid v \in \mathbb{N} \land a_v \in \mathbb{R} \}$
- A set of sensors $S \coloneqq \{ s_1,\dots,s_w \mid w \in \mathbb{N} \land s_w \in \mathbb{R} \}$
- Some kind of user interface
- A user (and a developer)


a1
d1
d2
a2
a3
UI
H
d3
s1
d4
s2
direct Interaction
indirect Interaction
Component


UI
Smart Home Hub
a_1
a_2
s_2
Goal: 600 lm
execution I
set(d_1, 30%)
set(d_1, 30%)
set(a_1, 30%)
set(d_2, 100%)
set(d_2, 100%)
set(a_2, 100%)
validation I
get(d_4)
get(d_4)
get(s_2)
450 lm
450 lm
450 lm
execution II
set(d_1, 60%)
set(d_1, 60%)
set(a_1, 60%)
validation II
get(d_4)
get(d_4)
get(s_2)
600 lm
600 lm
600 lm

Users usually know what they want to do, but they do not know how to do it.

[1]



RQ 1

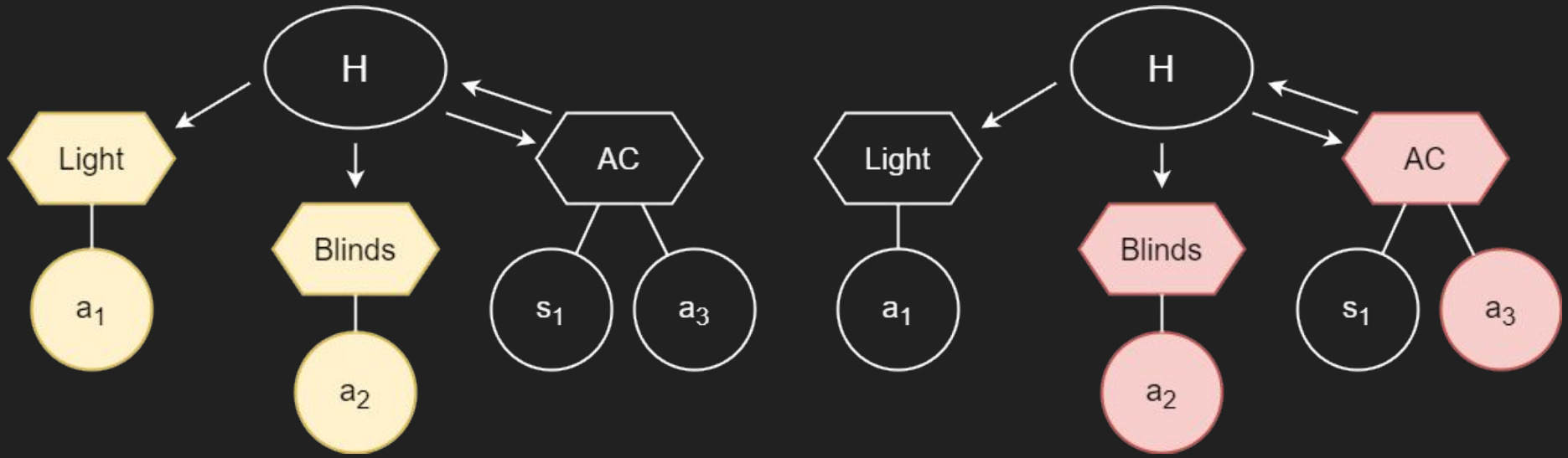
How to deal with and connect to a wide variety of smart home systems?

RQ 2

How to deal with and allow incorporation of different prediction models?

RQ 3

Which software qualities are crucial for developing a sustainable and useful solution? What can be done to accomplish and enforce selected software qualities?


H
Light
AC
Blinds
a_1
a_2
s_1
a_3
H
Light
AC
Blinds
a_1
a_2
s_1
a_3

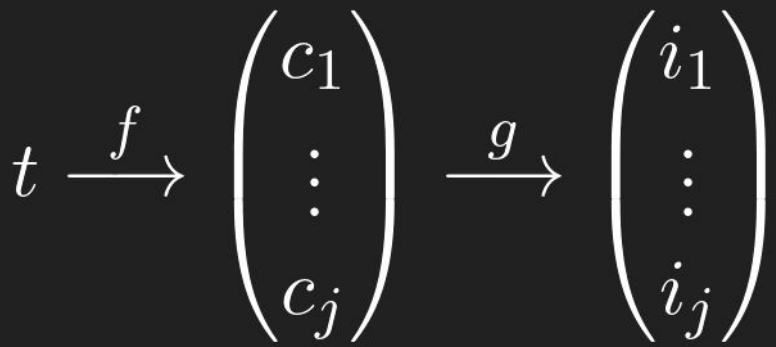
A set of environment variables $V \coloneqq \{ v_1,\dots,v_i \mid i \in \mathbb{N} \}$

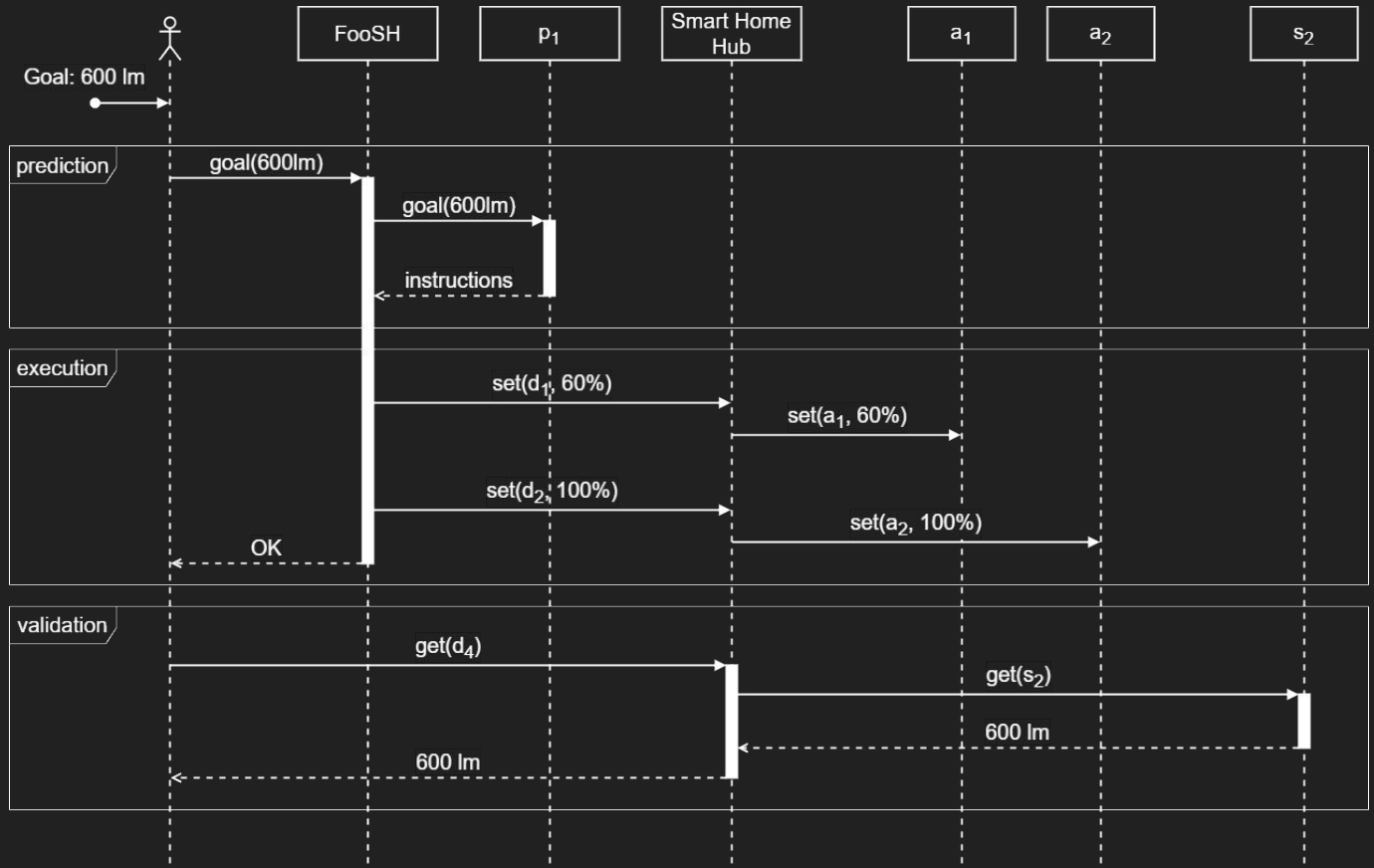
$v_i \subseteq D, v_i \in V$

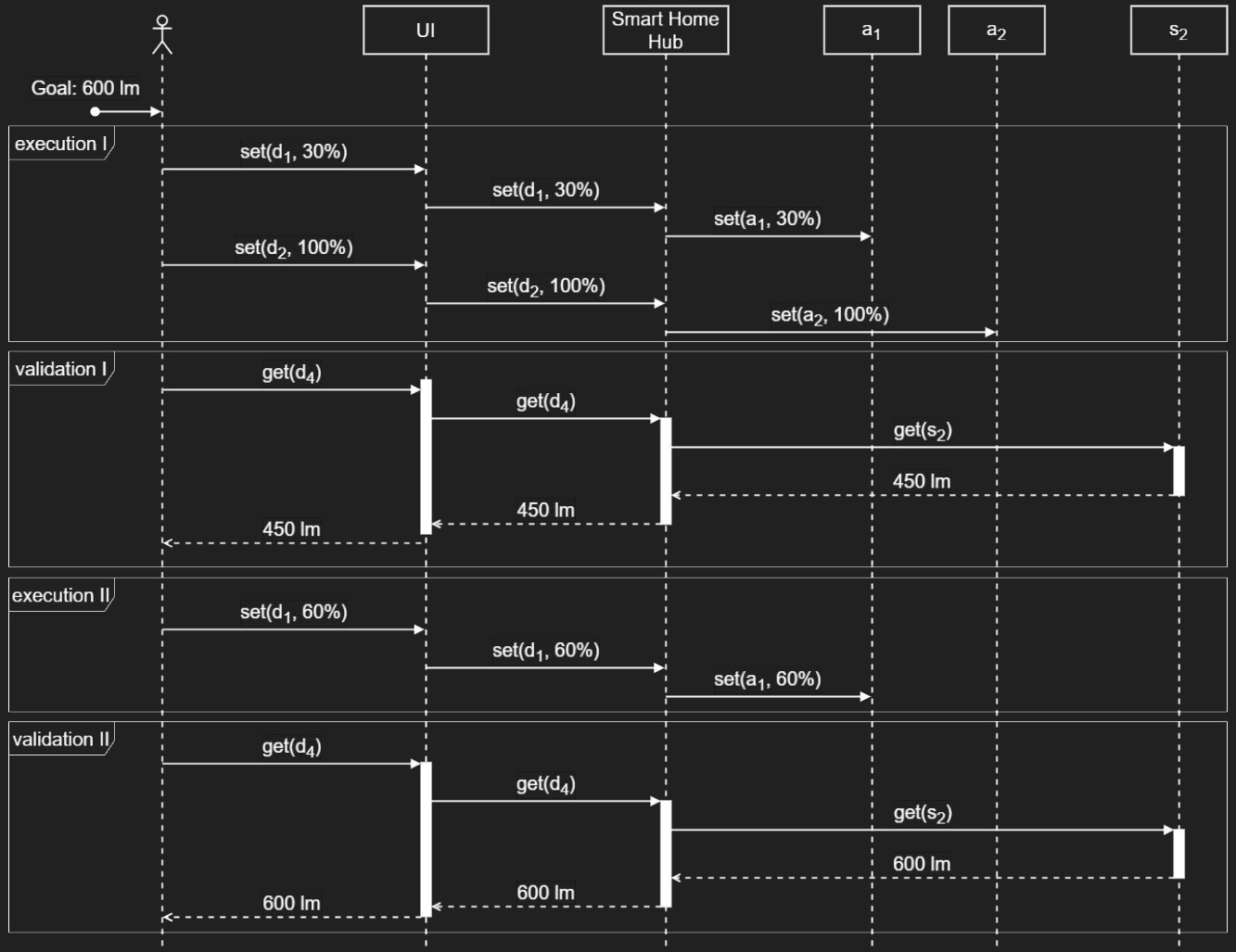
A set of prediction models $P \coloneqq \{ (T, LV, f, g) \mid LV \subseteq V \}$

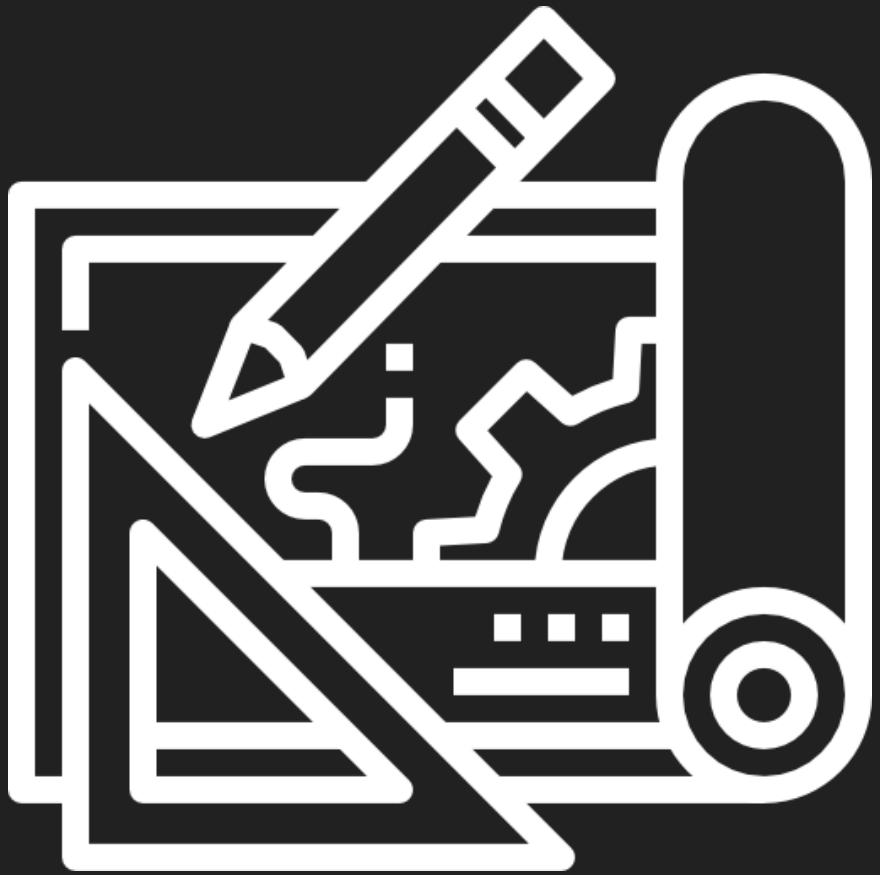
A set of prediction models $P \coloneqq \{ (T, LV, f, g) \mid LV \subseteq V \}$

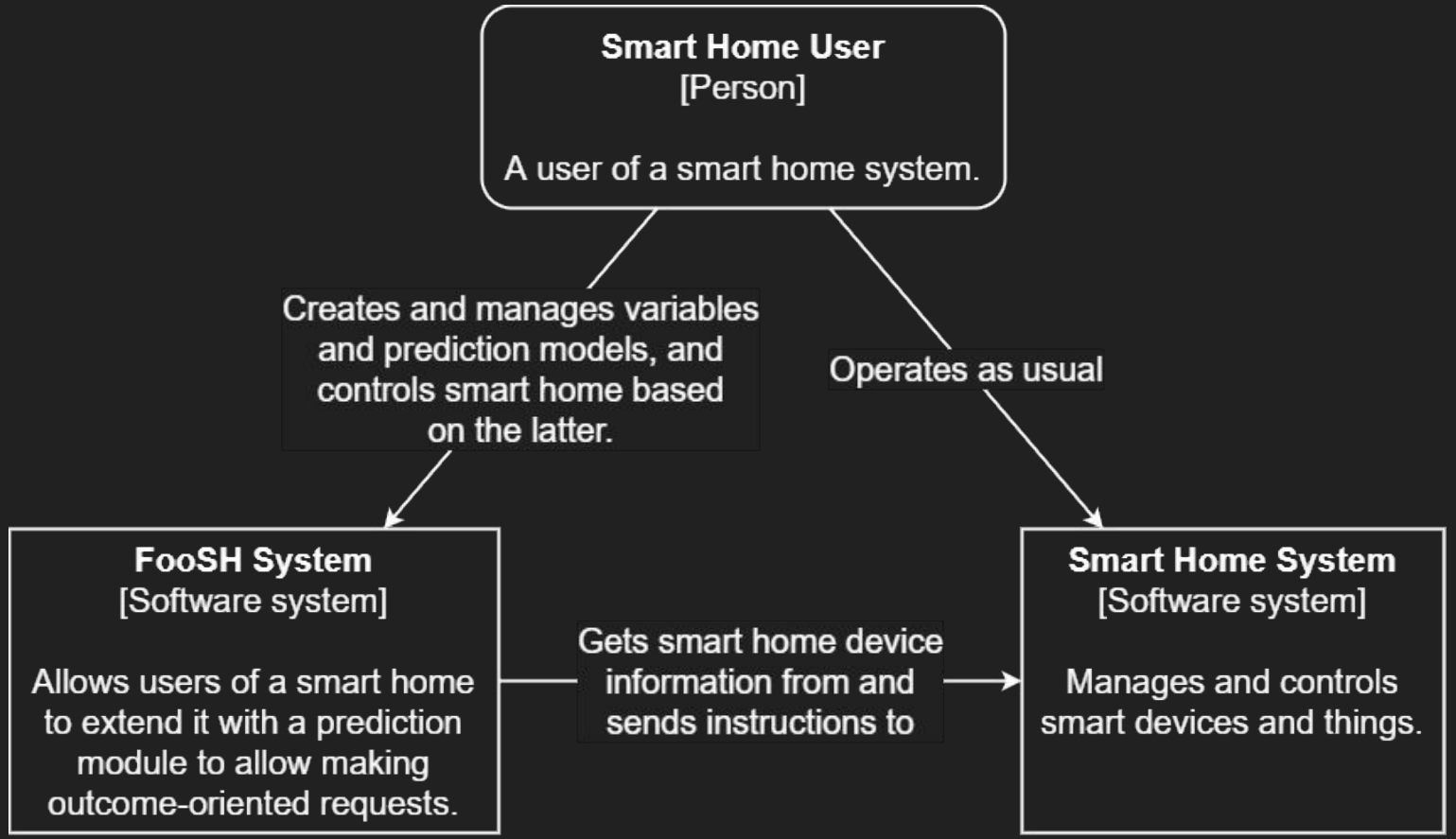
- Target space T
- Linked variables LV
- Prediction function f
- Translation function g



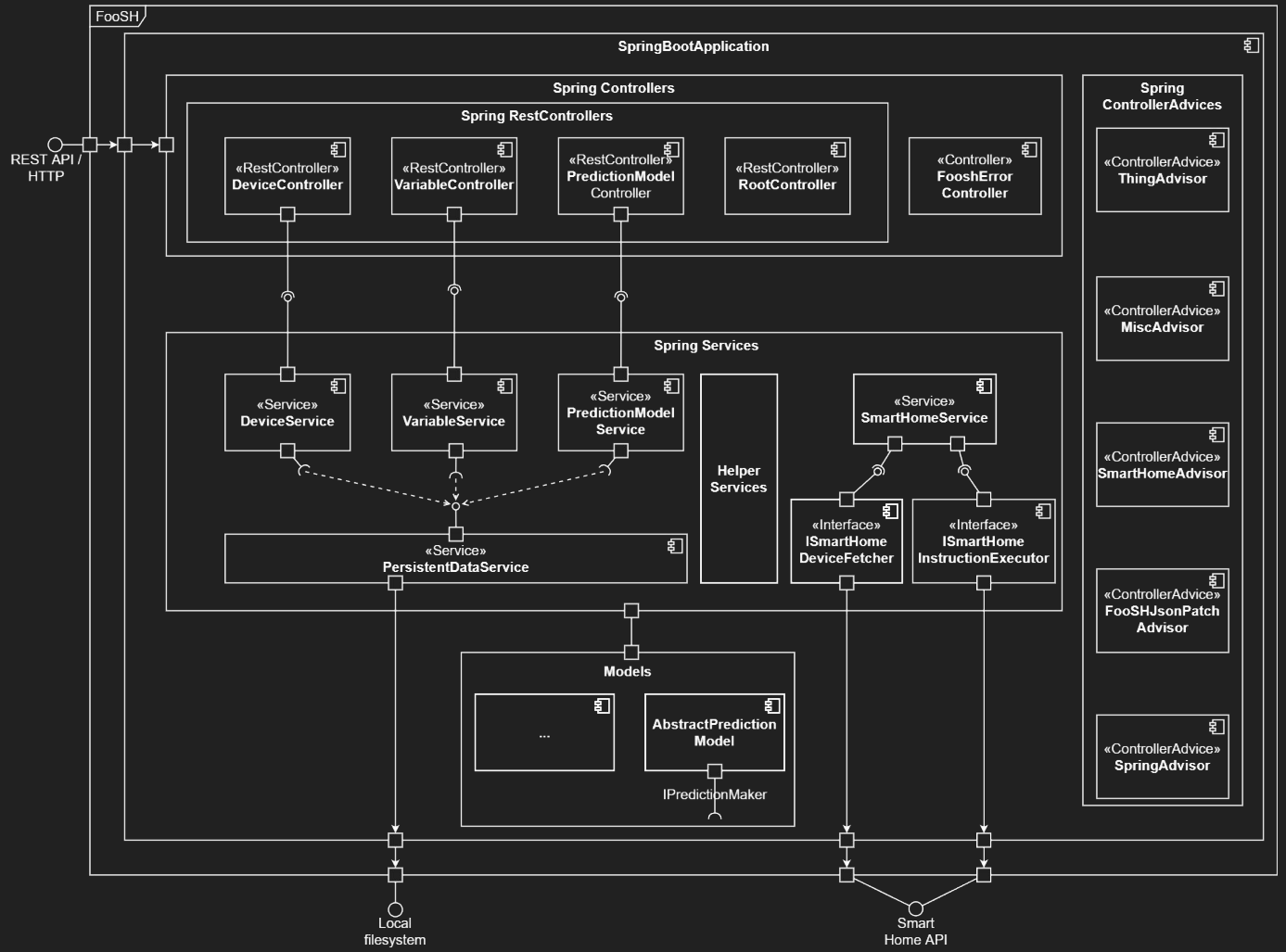

Goal: 600 lm
FooSH
p_1
Smart Home Hub
a_1
a_2
s_2
prediction
goal(600lm)
goal(600lm)
instructions
execution
set(d_1, 60%)
set(a_1, 60%)
set(d_2, 100%)
set(a_2, 100%)
OK
validation
get(d_4)
get(s_2)
600 lm
600 lm


UI
Smart Home Hub
a_1
a_2
s_2
Goal: 600 lm
execution I
set(d_1, 30%)
set(d_1, 30%)
set(a_1, 30%)
set(d_2, 100%)
set(d_2, 100%)
set(a_2, 100%)
validation I
get(d_4)
get(d_4)
get(s_2)
450 lm
450 lm
450 lm
execution II
set(d_1, 60%)
set(d_1, 60%)
set(a_1, 60%)
validation II
get(d_4)
get(d_4)
get(s_2)
600 lm
600 lm
600 lm




Smart Home User
[Person]
A user of a smart home system.
Creates and manages variables and prediction models, and controls smart home based on the latter.
Operates as usual
FooSH System
[Software system]
Allows users of a smart home to extend it with a prediction module to allow making outcome-oriented requests.
Gets smart home device information from and sends instructions to
Smart Home System
[Software system]
Manages and controls smart devices and things.


REST
API


FooSH
SpringBootApplication
REST API / HTTP
Spring Controllers
Spring RestControllers
«RestController» DeviceController
«RestController» VariableController
«RestController» PredictionModel Controller
«RestController» RootController
«Controller» FooshError Controller
Spring ControllerAdvices
«ControllerAdvice» ThingAdvisor
«ControllerAdvice» MiscAdvisor
«ControllerAdvice» SmartHomeAdvisor
«ControllerAdvice» FooSHJsonPatch Advisor
«ControllerAdvice» SpringAdvisor
Spring Services
«Service» DeviceService
«Service» VariableService
«Service» PredictionModel Service
Helper Services
«Service» SmartHomeService
«Interface» ISmartHome DeviceFetcher
«Interface» ISmartHome InstructionExecutor
«Service» PersistentDataService
Models
...
AbstractPrediction Model
IPredictionMaker
Local filesystem
Smart Home API

How to operate FooSH?

aka. The 5 Steps to Success!

Step 1: Fetch smart devices

```
POST /api/devices/ HTTP/1.1
Accept: application/json

{
    ”details”: {
        ”token”: ”abc123”,
        ”user”: ”foo”,
        ”secret”: ”bar”,
    }
}
```

Step 2: Define environment variable

```
POST /api/vars/ HTTP/1.1
Accept: application/json

{
    ”name”: ”brightness”
}
```

Step 3: Assign device(s) to variable

```
POST /api/vars/brightness/devices/ HTTP/1.1
Accept: application/json

{
   ”deviceIds”: [
      ”light1”,
      ”light2”
   ]
}
```

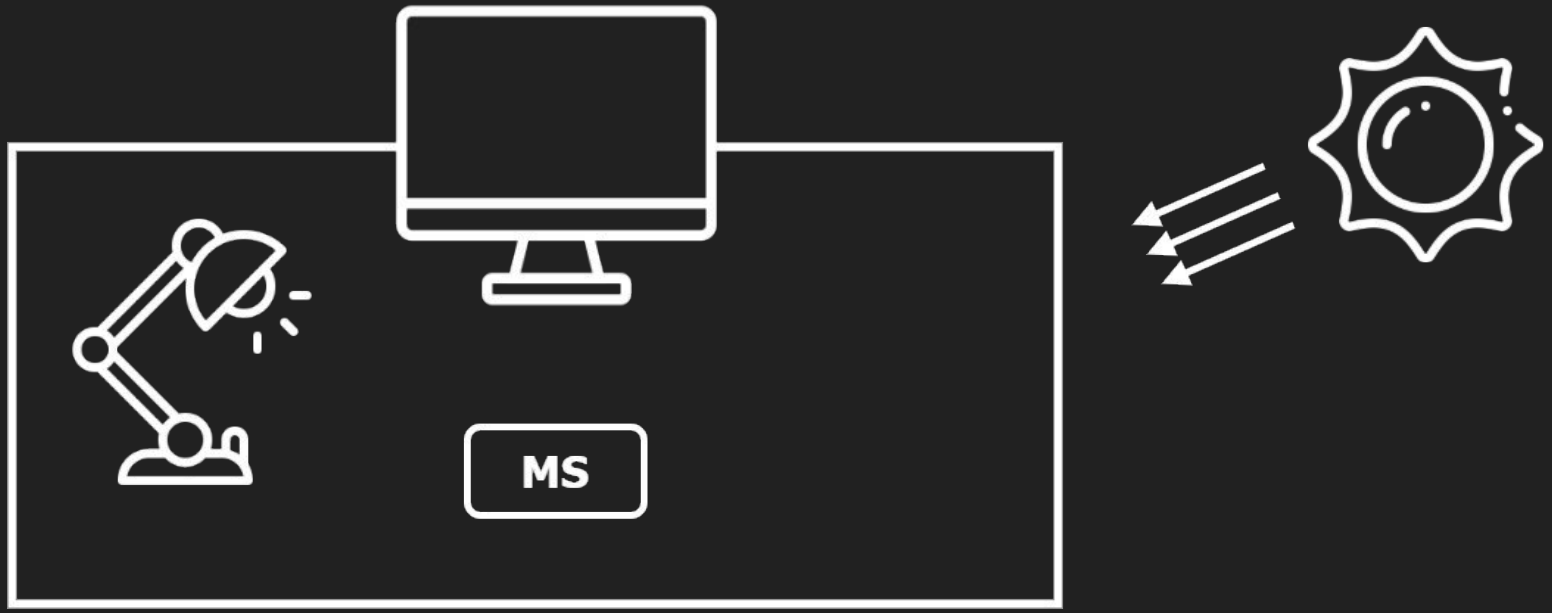
Step 4: Link variable with prediction model

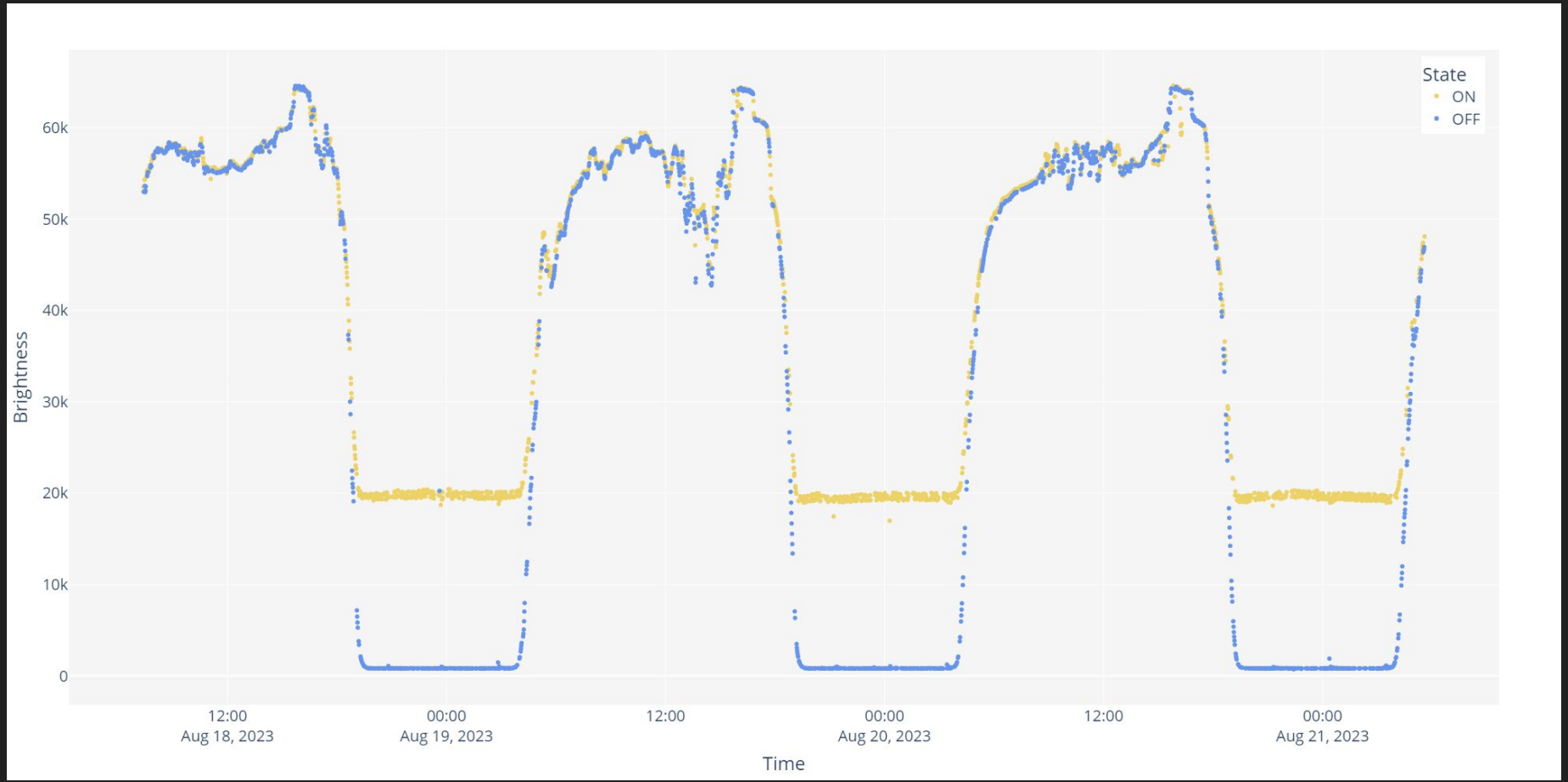
```
POST /api/models/my-model/mappings/ HTTP/1.1
Accept: application/json

{
    ”variableId”: ”brightness”,
    ”mappings”: [
        {
            ”parameter”: ”x1”,
            ”deviceId”: ”light1”
        },
        {
            ”parameter”: ”x2”,
            ”deviceId”: ”light2”
        }
    ]
}
```

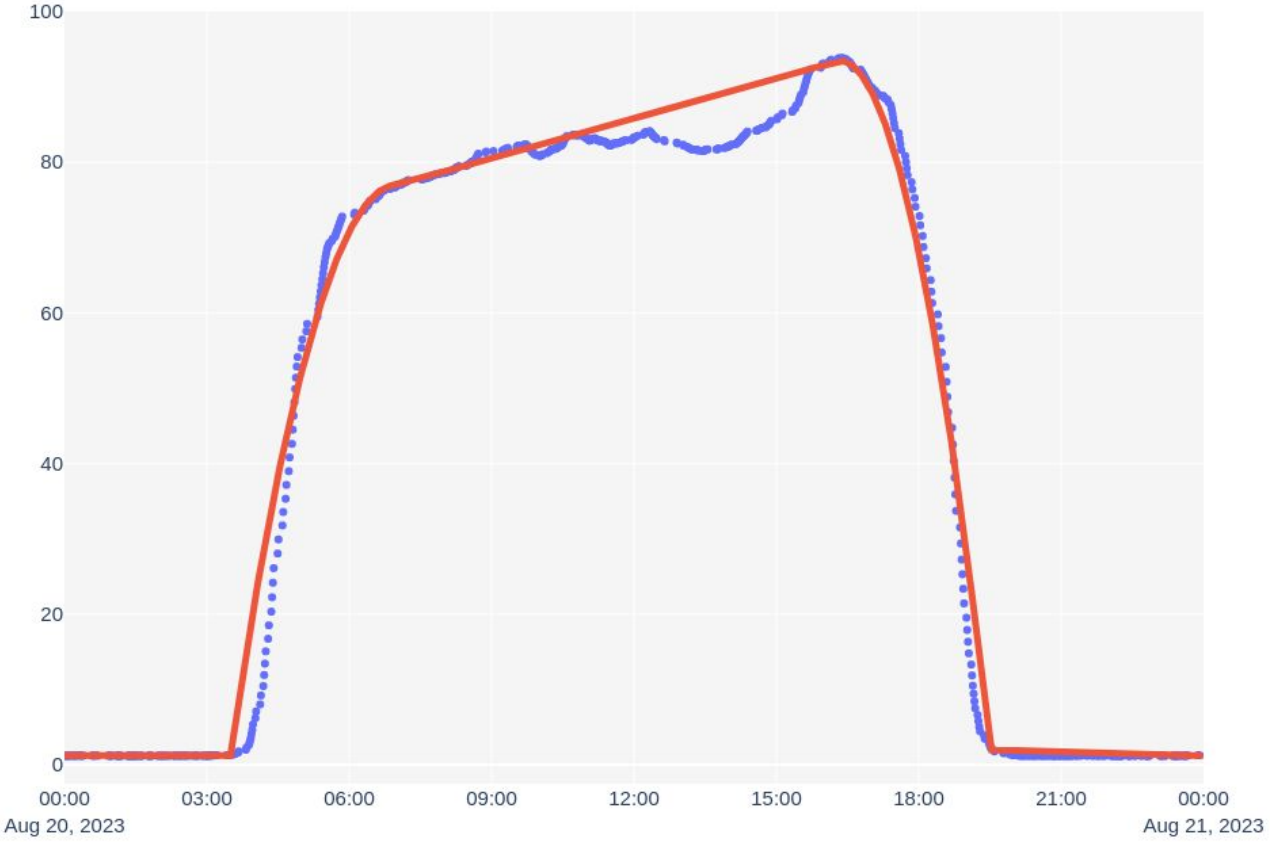
Step 5: Make a prediction

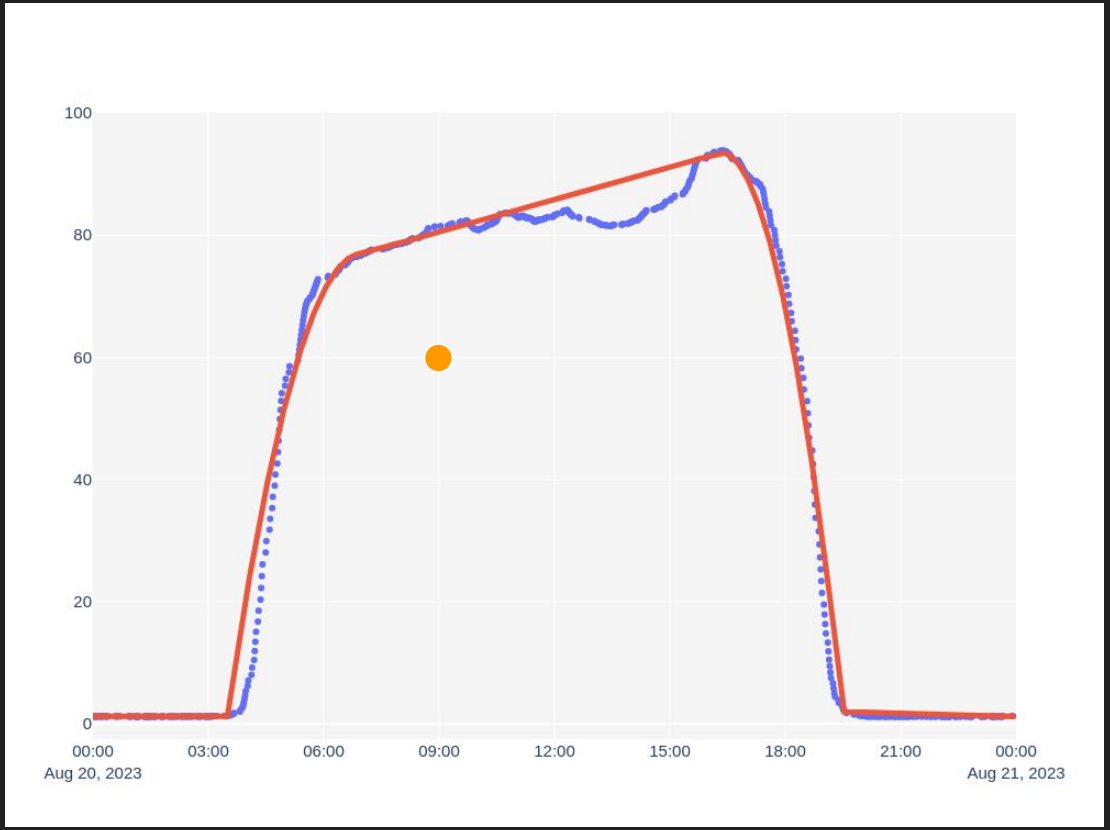
```
POST /api/vars/brightness HTTP/1.1
Accept: application/json

{
    ”modelId”: ”my-model”,
    ”value”: 50,
    ”execute”: true
}
```

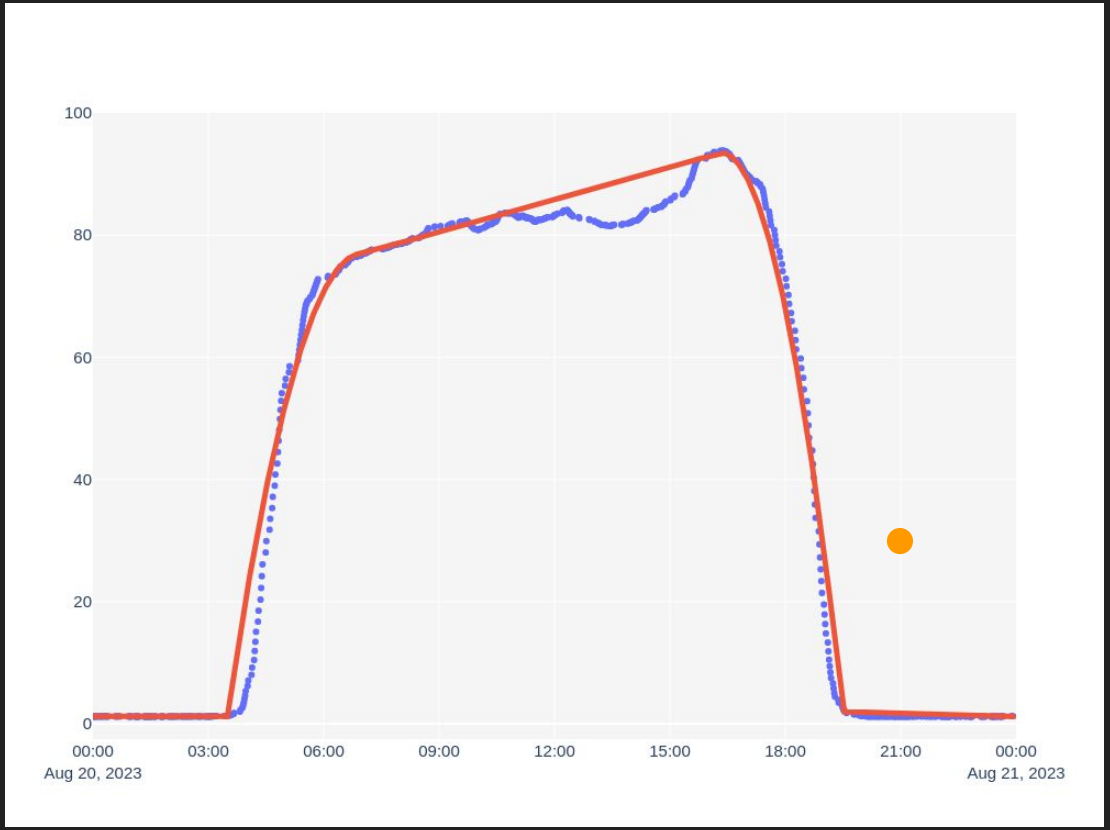
Proof of Concept


MS


State
ON
OFF
Brightness
60k
50k
40k
30k
20k
10k
0
12:00
Aug 18, 2023
00:00
Aug 19, 2023
12:00
00:00
Aug 20, 2023
12:00
00:00
Aug 21, 2023
Time


100
80
60
40
20
0
00:00
Aug 20, 2023
03:00
06:00
09:00
12:00
15:00
18:00
21:00
00:00
Aug 21, 2023


100
80
60
40
20
0
00:00
Aug 20, 2023
03:00
06:00
09:00
12:00
15:00
18:00
21:00
00:00
Aug 21, 2023

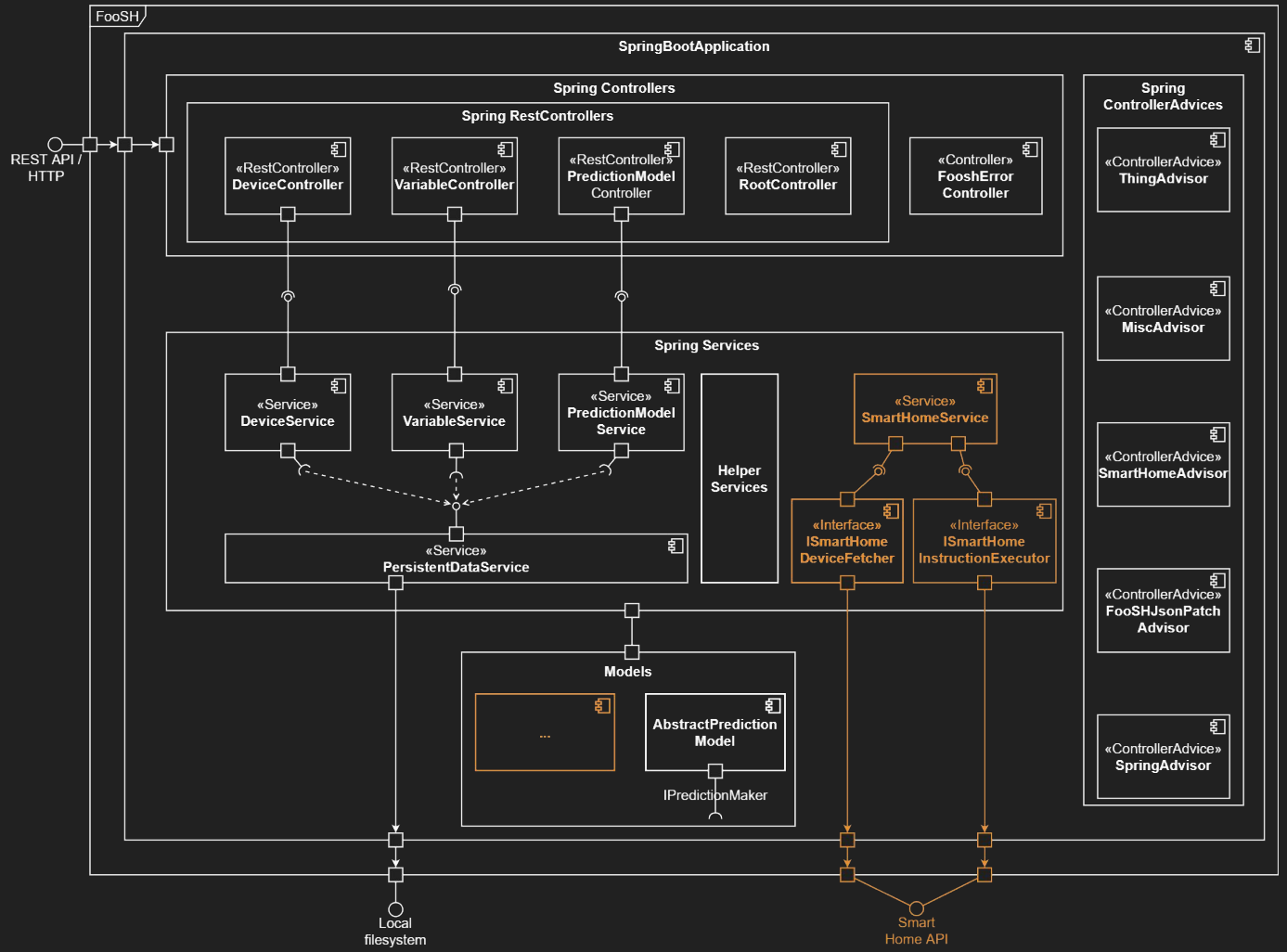

100
80
60
40
20
0
00:00
Aug 20, 2023
03:00
06:00
09:00
12:00
15:00
18:00
21:00
00:00
Aug 21, 2023

Results

- Validation by Trial-and-Error
- Prediction Model
  - MAE : 3.195
  - MSE : 29.422
  - MSLE: 0.081

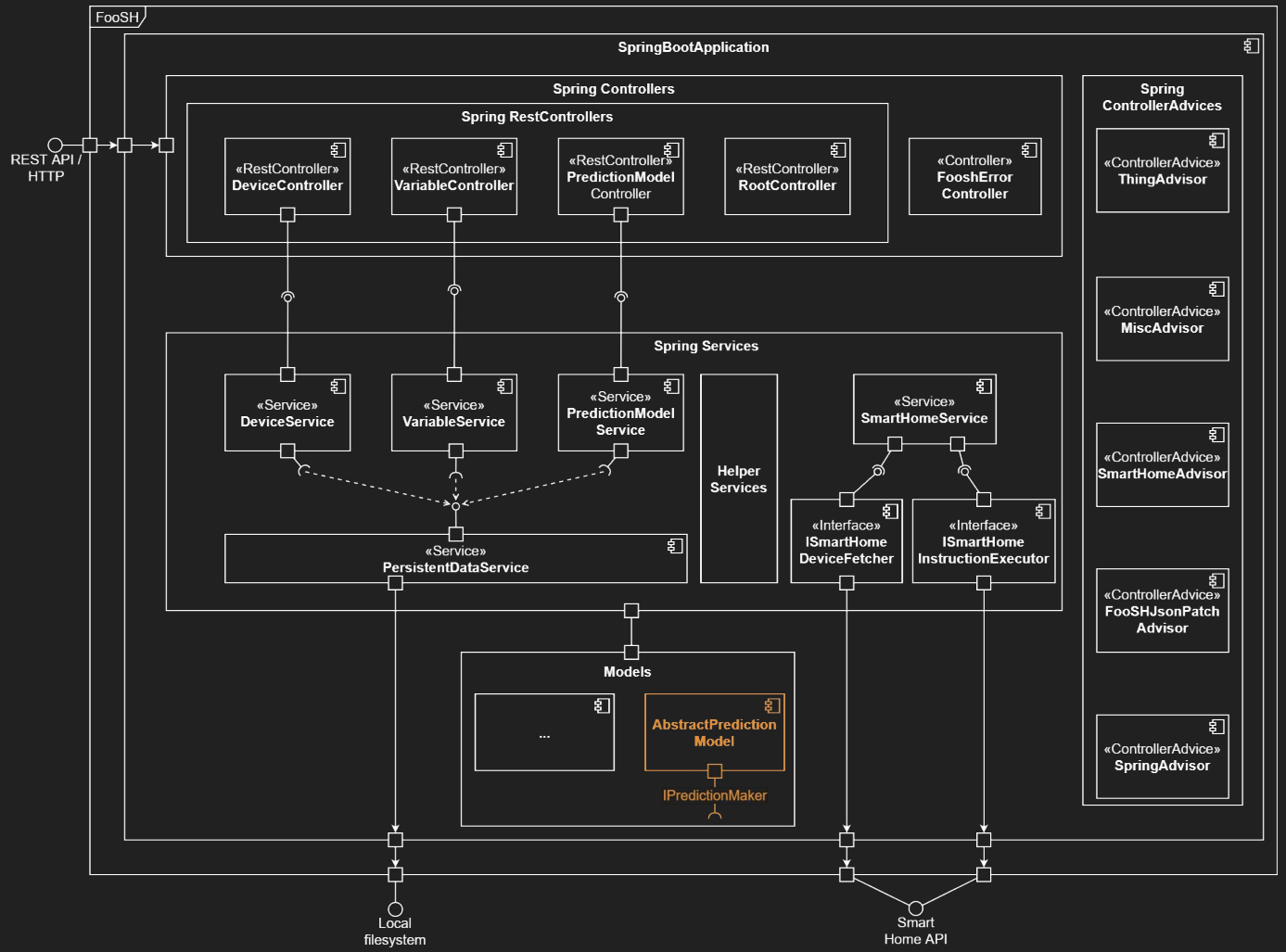
RQ 1

How to deal with and connect to a wide variety of smart home systems?


FooSH
SpringBootApplication
REST API / HTTP
Spring Controllers
Spring RestControllers
«RestController» DeviceController
«RestController» VariableController
«RestController» PredictionModel Controller
«RestController» RootController
«Controller» FooshError Controller
Spring ControllerAdvices
«ControllerAdvice» ThingAdvisor
«ControllerAdvice» MiscAdvisor
«ControllerAdvice» SmartHomeAdvisor
«ControllerAdvice» FooSHJsonPatch Advisor
«ControllerAdvice» SpringAdvisor
Spring Services
«Service» DeviceService
«Service» VariableService
«Service» PredictionModel Service
Helper Services
«Service» SmartHomeService
«Interface» ISmartHome DeviceFetcher
«Interface» ISmartHome InstructionExecutor
«Service» PersistentDataService
Models
...
AbstractPrediction Model
IPredictionMaker
Local filesystem
Smart Home API

RQ 2

How to deal with and allow incorporation of different prediction models?


FooSH
SpringBootApplication
REST API / HTTP
Spring Controllers
Spring RestControllers
«RestController» DeviceController
«RestController» VariableController
«RestController» PredictionModel Controller
«RestController» RootController
«Controller» FooshError Controller
Spring ControllerAdvices
«ControllerAdvice» ThingAdvisor
«ControllerAdvice» MiscAdvisor
«ControllerAdvice» SmartHomeAdvisor
«ControllerAdvice» FooSHJsonPatch Advisor
«ControllerAdvice» SpringAdvisor
Spring Services
«Service» DeviceService
«Service» VariableService
«Service» PredictionModel Service
Helper Services
«Service» SmartHomeService
«Interface» ISmartHome DeviceFetcher
«Interface» ISmartHome InstructionExecutor
«Service» PersistentDataService
Models
...
AbstractPrediction Model
IPredictionMaker
Local filesystem
Smart Home API

RQ 3

Which software qualities are crucial for developing a sustainable and useful solution? What can be done to accomplish and enforce selected software qualities?

ISO/IEC 25010

Systems and Software Quality Requirements and Evaluation (SQuaRE)

[5]

ISO/IEC 25010

Product Quality

- Functional Suitability
- Performance Efficiency
- Compatibility
- Usability
- Reliability
- Security
- Maintainability
- Portability

Quality in Use

- Effectiveness
- Efficiency
- Satisfaction
- Freedom of Risk
- Context Coverage

ISO/IEC 25010

Product Quality

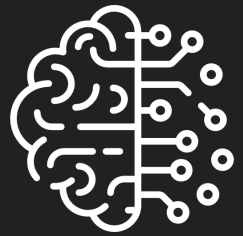
- Functional Suitability
- Performance Efficiency
- Compatibility
- Usability
- Reliability
- Security
- Maintainability
- Portability

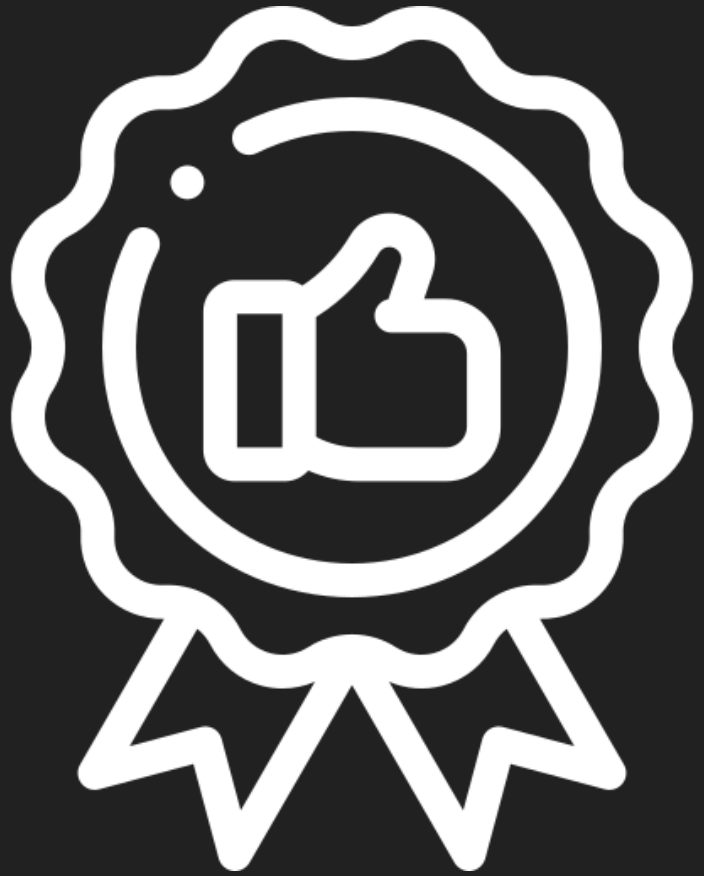
Quality in Use

- Effectiveness
- Efficiency
- Satisfaction
- Freedom of Risk
- Context Coverage


LIMITED


LIMITED





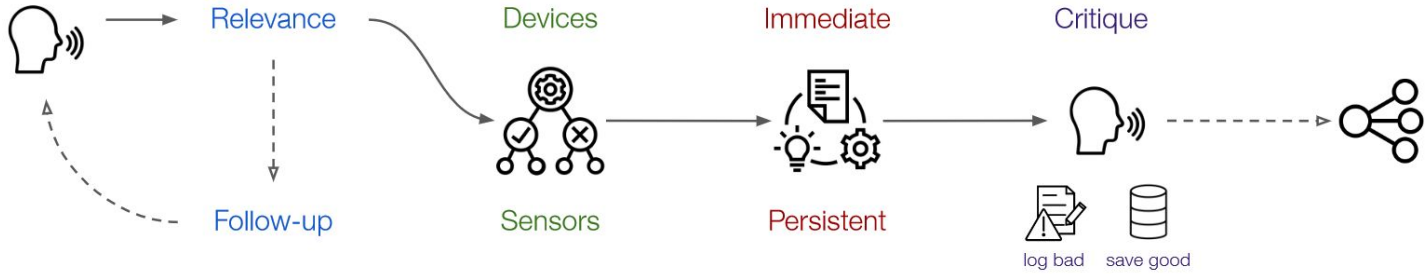
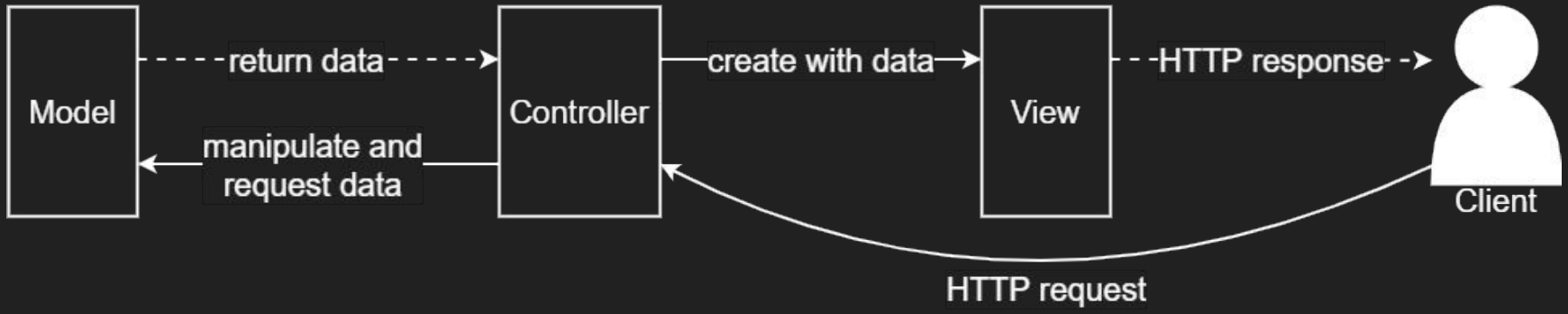


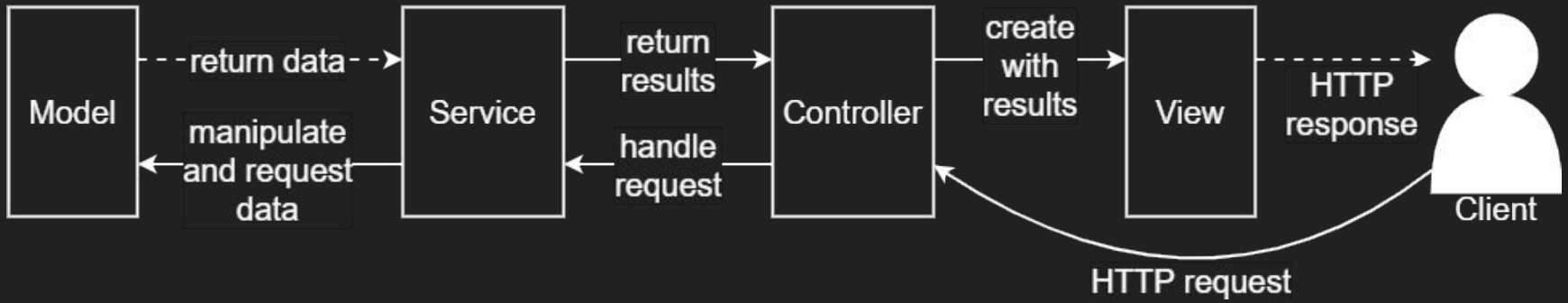
Fig. 15. Sasha uses iterative reasoning to generate consistently high-quality action plans that leverage whichever relevant devices are available in a given smart home.

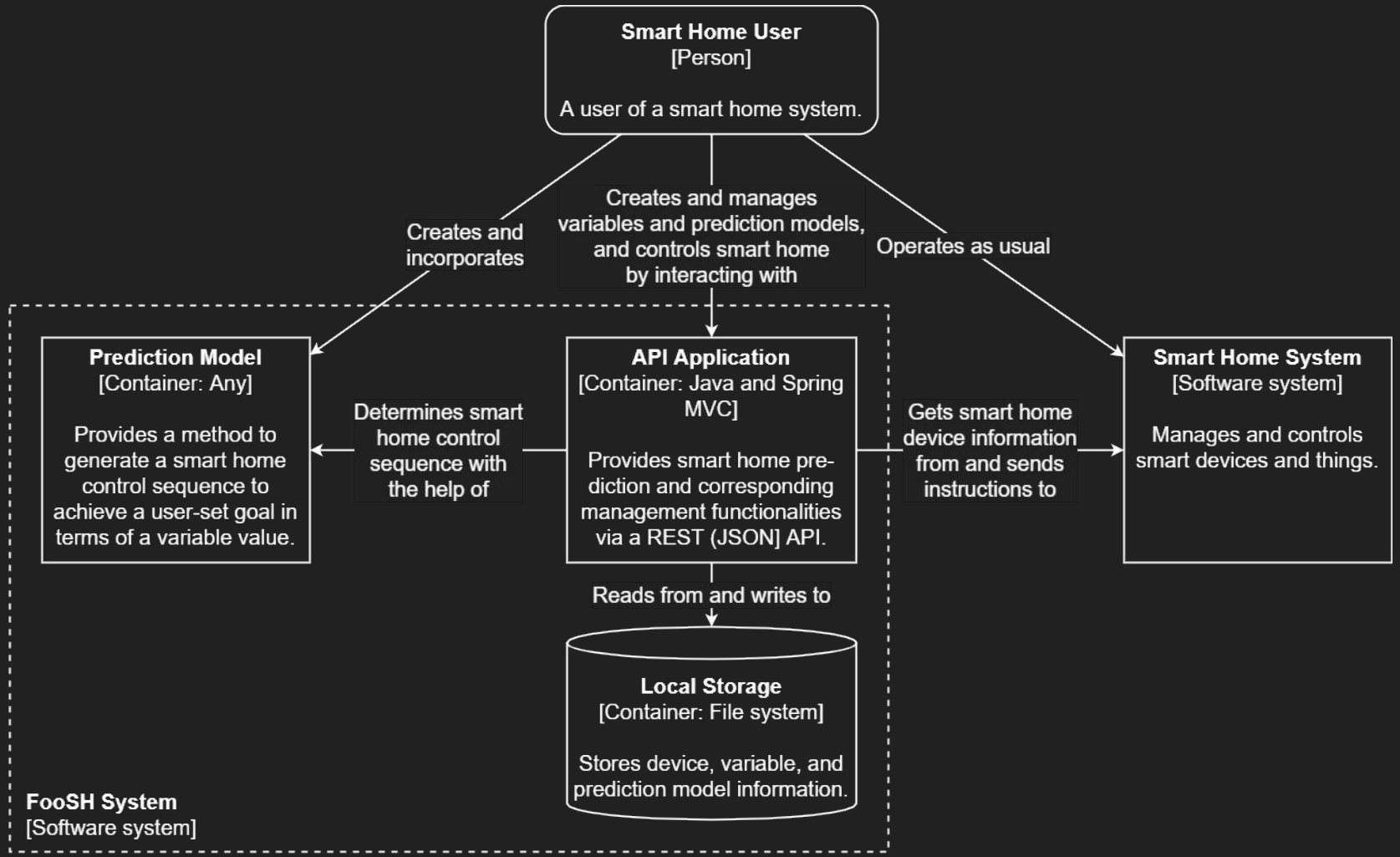
[3]

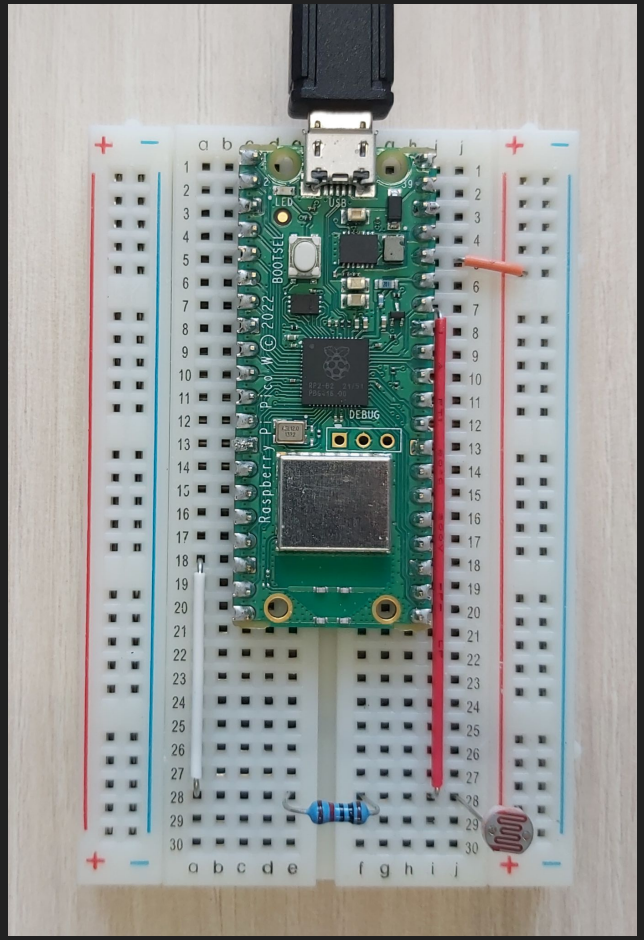
Sources

APPENDIX

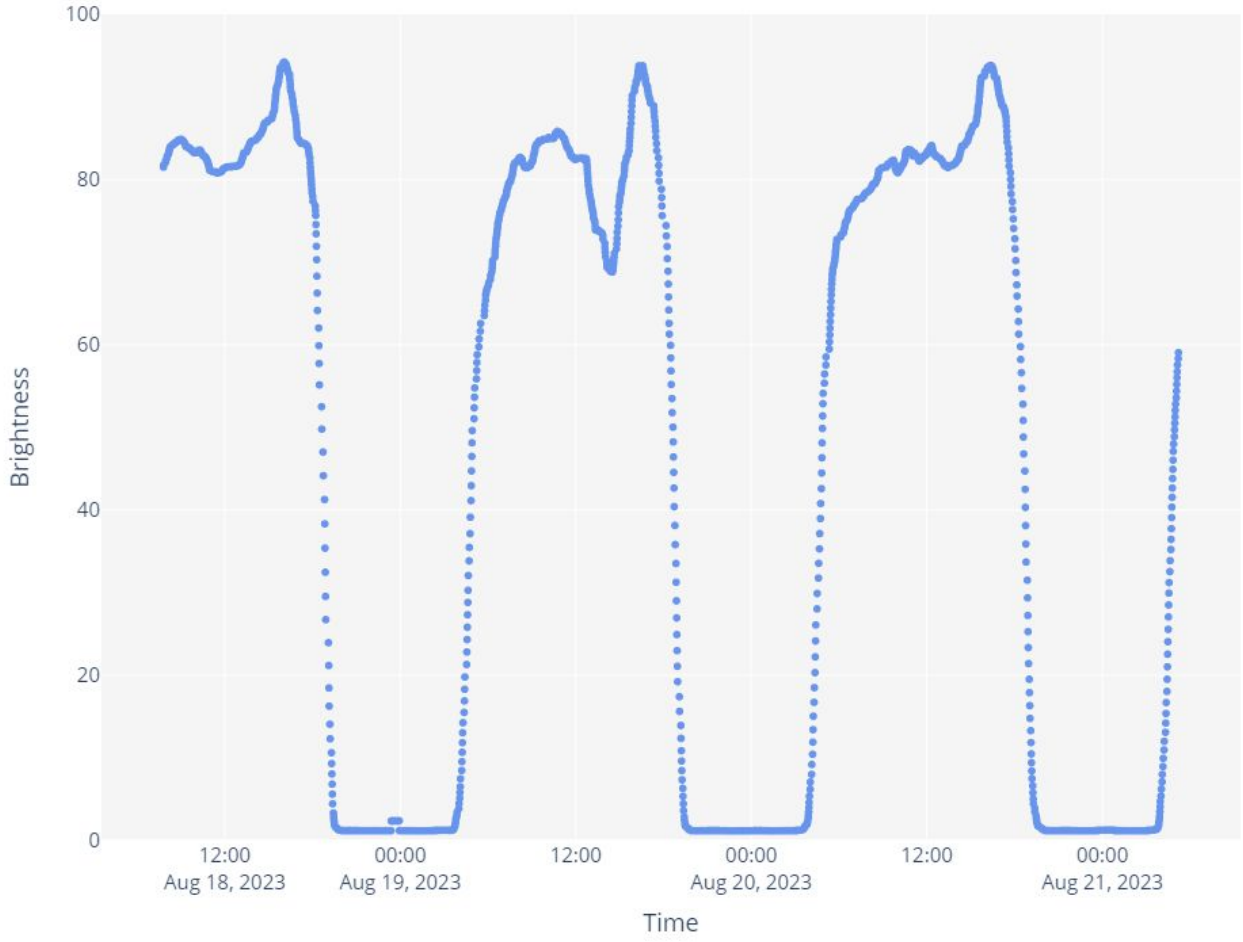

Model
return data
Controller
create with data
View
HTTP response
manipulate and request data
Client
HTTP request

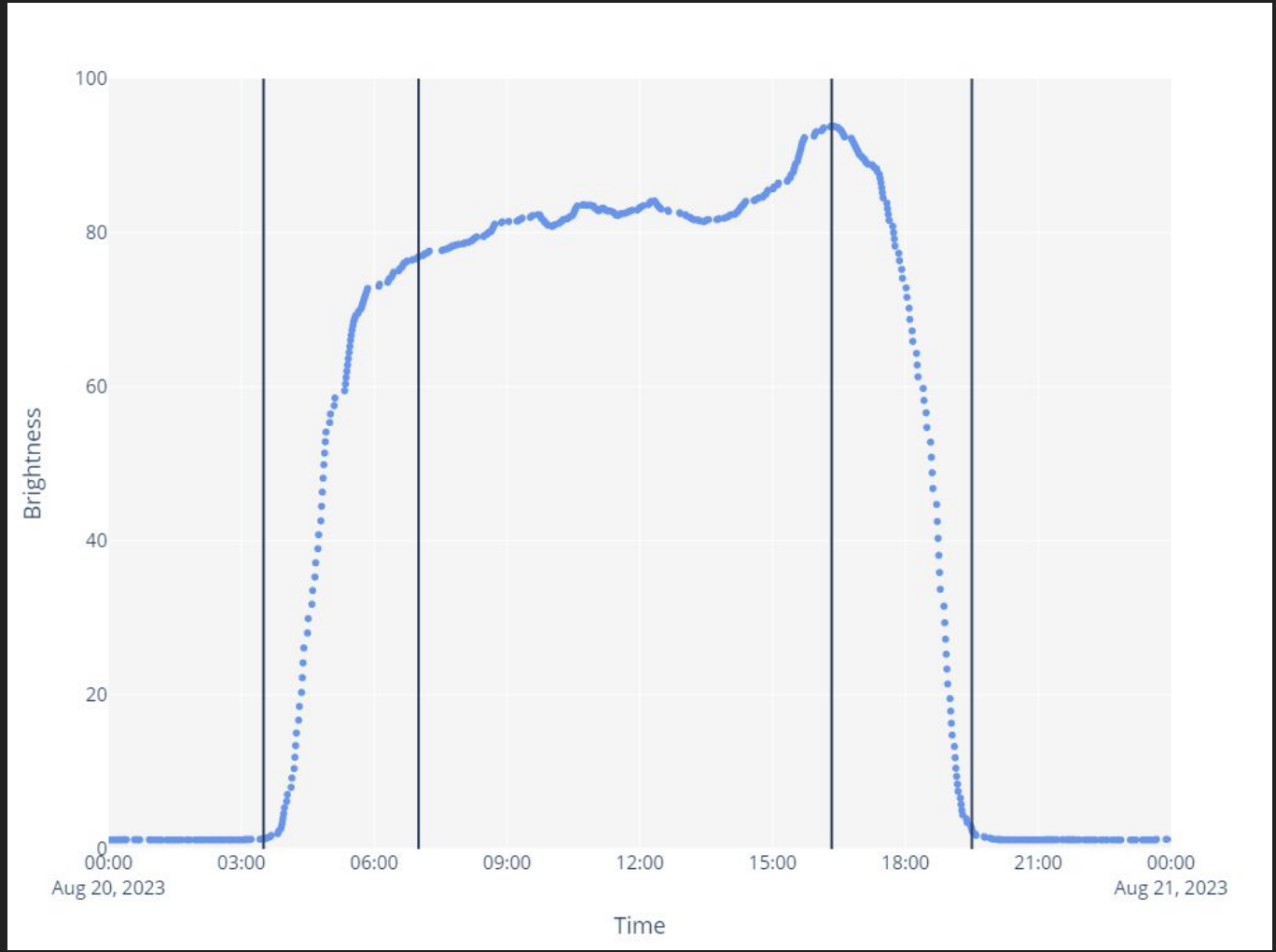

Model
return data
Service
return results
Controller
create with results
View
HTTP response
manipulate and request data
handle request
Client
HTTP request


Smart Home User
[Person]
A user of a smart home system.
Creates and incorporates
Creates and manages variables and prediction models, and controls smart home by interacting with
Operates as usual
Prediction Model
[Container: Any]
Provides a method to generate a smart home control sequence to achieve a user-set goal in terms of a variable value.
Determines smart home control sequence with the help of
API Application
[Container: Java and Spring MVC]
Provides smart home prediction and corresponding management functionalities via a REST (JSON) API.
Gets smart home device information from and sends instructions to
Smart Home System
[Software system]
Manages and controls smart devices and things.
Reads from and writes to
Local Storage
[Container: File system]
Stores device, variable, and prediction model information.
FooSH System
[Software system]


Raspberry Pi Pico W © 2022
BOOTSEL
LED
USB
DEBUG

```
{
     "timestamp":  1692343421703,
     "value": 53517,
     "items": [
          {
               "link": "http://192.168.108.103:8080/rest/items/plug1 ",
               "state": "ON",
               "editable": true,
               "type": "Switch",
               "name": "plug1",
               "label": "shelly-plug-1",
               "tags": ["Office"],
               "groupNames": ["Office"]
          },
          {
               "link": "http://192.168.108.103:8080/rest/items/foo ",
               ...
               "groupNames": ["Office"]
          },
          ...
     ]
}
```


100
80
60
40
20
0
Brightness
12:00
Aug 18, 2023
00:00
Aug 19, 2023
12:00
00:00
Aug 20, 2023
12:00
00:00
Aug 21, 2023
Time


100
80
60
40
20
0
Brightness
00:00
Aug 20, 2023
03:00
06:00
09:00
12:00
15:00
18:00
21:00
00:00
Aug 21, 2023
Time

```
procedure makePrediction(targetValue : v)
    instr ← [ ]
    t ← time

    if f(t) ≤ v then
        instr ← instr + turn_light_off
    else
        instr ← instr + turn_light_on
    end if


    return instr
end procedure
```

ISO/IEC 25010

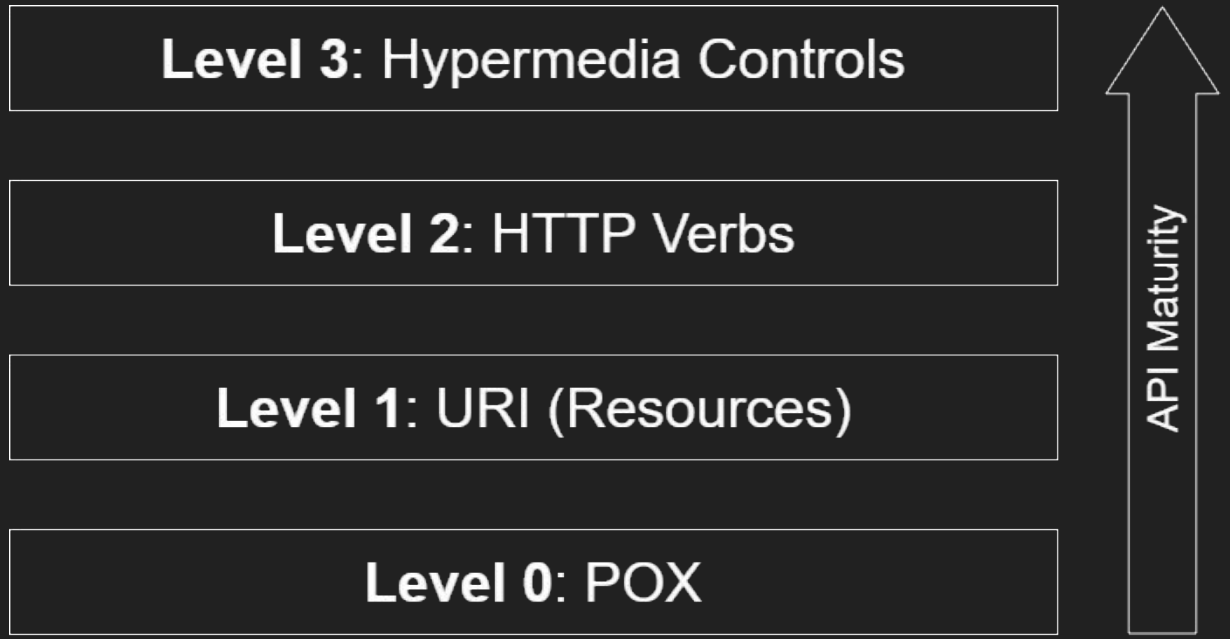
Product Quality

- Functional Suitability
  - Fct. Completeness
  - Fct. Correctness
  - Fct. Appropriateness
- Performance Efficiency
  - Time-behavior
  - Resource Utilization
  - Capacity
- Compatibility
  - Co-existence
  - Interoperability
- Usability
  - Appropriateness Recognizability
  - Learnability
  - Operability
  - User Error Protection
  - User Interface Aesthetics
  - Accessibility
- Reliability
  - Maturity
  - Availability
  - Fault Tolerance
  - Recoverability
- Security
  - Confidentiality
  - Integrity
  - Non-repudiation
  - Accountability
  - Authenticity
- Maintainability
  - Modularity
  - Reusability
  - Analyzability
  - Modifiability
  - Testability
- Portability
  - Adaptability
  - Installability
  - Replaceability

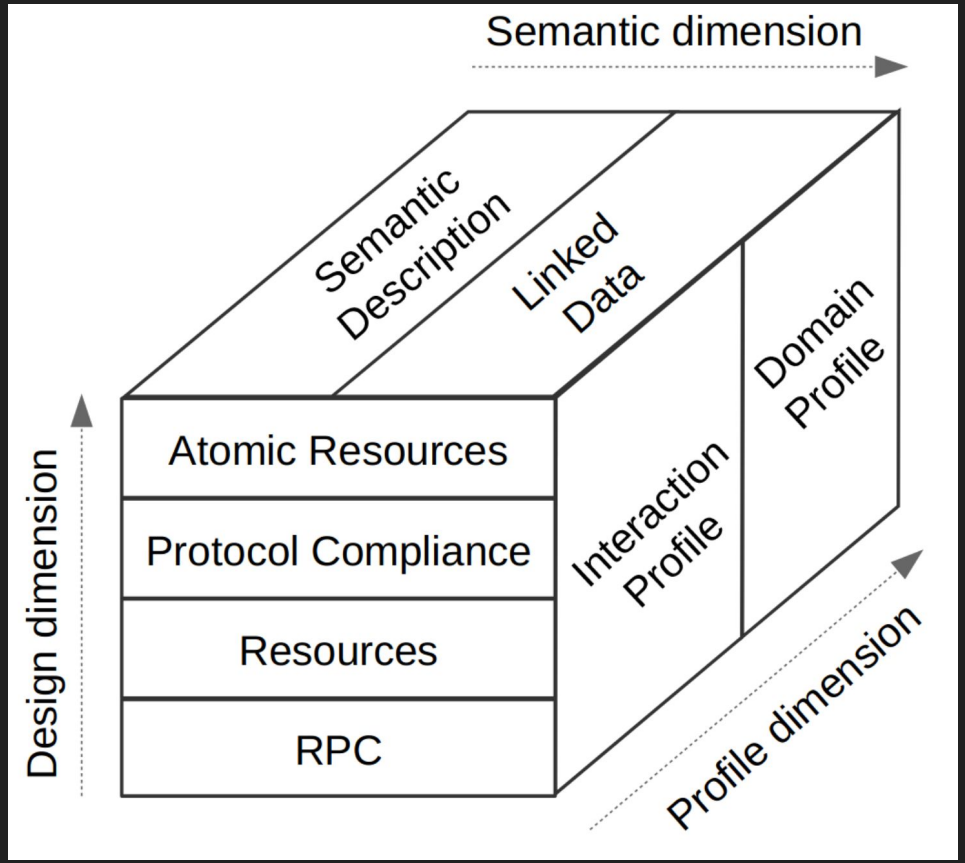
ISO/IEC 25010

Quality in Use

- Effectiveness
- Efficiency
- Satisfaction
  - Usefulness
  - Trust
  - Pleasure
- Freedom of Risk
  - Economic Risk Mitigation
  - Health and Safety Risk Mitigation
  - Env. Risk Mitigation
- Context Coverage
  - Context Completeness
  - Flexibility


Level 3: Hypermedia Controls
Level 2: HTTP Verbs
Level 1: URI (Resources)
Level 0: POX
API Maturity

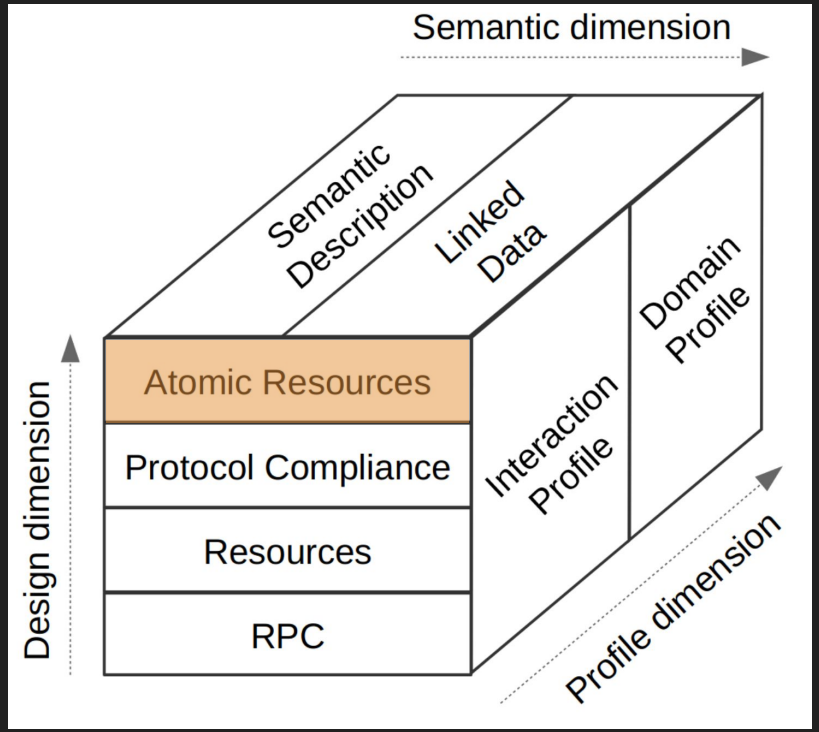
[4]


Semantic dimension
Semantic Description
Linked Data
Domain Profile
Atomic Resources
Protocol Compliance
Resources
RPC
Interaction Profile
Design dimension
Profile dimension

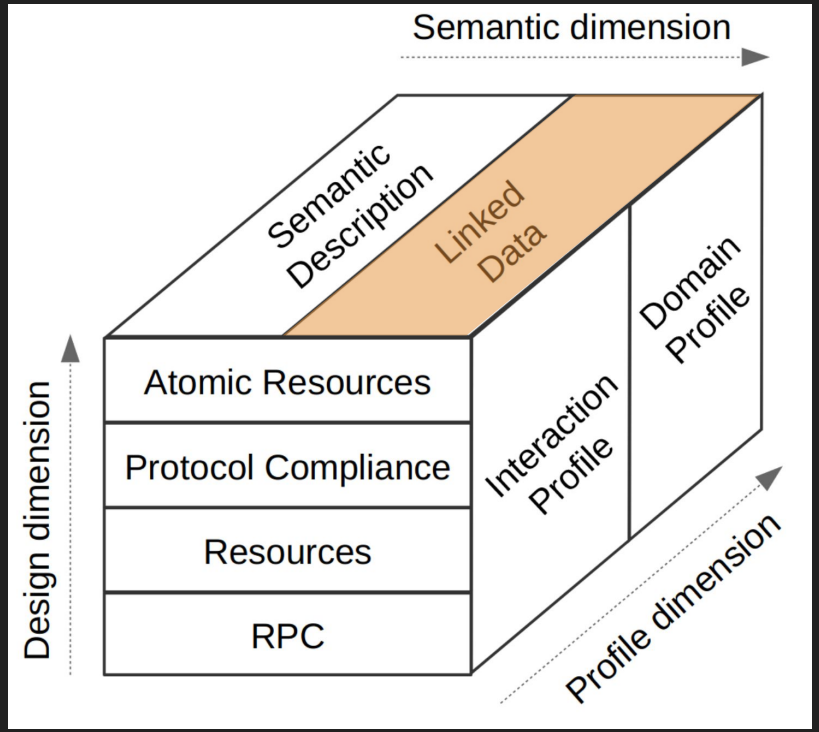
[5]

D4 - S2 - P1

Design: 4 - S2 - P1



D4 - Semantic: 2 - P1



D4 - S2 - Profile: 1