Smart Home Management

System Description Document

Contents

[1. Intro 2](#_Toc125660347)

[2. Technology 2](#_Toc125660348)

[3. Key Functionalities 2](#_Toc125660349)

[1. Configure the devices 2](#_Toc125660350)

[2. Collect data periodically 2](#_Toc125660351)

[3. Troubleshoot and control the devices 2](#_Toc125660352)

[4. Visualize the device details and provide some metrics 2](#_Toc125660353)

[4. Logical View 3](#_Toc125660354)

[5. Use Cases 3](#_Toc125660355)

[1. As an Admin user, I should be able to configure/install the SHM at a premise. 3](#_Toc125660356)

[2. As a General User, I should be able to control my appliances using the SHM client. 3](#_Toc125660357)

[3. As a General User, I should be able to view & monitor my appliances using SHM Client. 3](#_Toc125660358)

[6. Use Cases Detailed View 4](#_Toc125660359)

[1. As an Admin, I should be able to install and activate the SHM on premise. 4](#_Toc125660360)

[2. As a General user, I should be able to control the appliances. 5](#_Toc125660361)

[3. As a General user, I can view and monitor the appliances in my SHM client. 6](#_Toc125660362)

[7. Information model 6](#_Toc125660363)

[8. Security 10](#_Toc125660364)

[9. Wireframe UI Mockups 11](#_Toc125660365)

[10. Technology choice 12](#_Toc125660366)

History

|  |  |  |
| --- | --- | --- |
| **Owner** | **Changes** | **Date/time** |
| Srivatsan M | Created | 23/01/2023 |
| Srivatsan M | Completed Initial version | 24/01/2023 |
| Srivatsan M | Enhanced the YANG model, updated block diagram, Added Wireframe UI mocks and Technology choice sections | 26/01/2023 |

# Intro

Require a smart home management solution which can manage all these smart devices via NETCONF/YANG. Each of the device has some common characteristics and some specific characteristics. To start with, this document describes about home management of Fridge & Air conditioner. We would refer to Smart Home Management to as SHM in some parts of this document.

The SHM app via Smartphone or via website would ensure the polling of device information and storing them in the cloud storage for later analytics and also for user preferences management.

# Technology

The choice of technology for configuring, managing & monitoring the smart home appliance is NETCONF protocol via the YANG specification due to its proven benefits of network element management capabilities. The system can be evolved further with RESTCONF and / or other methods in the future.

Refer Section:10 (Technology Choice) for more info.

# Key Functionalities

## Configure the devices

This function helps in configuring the appliances managed by SHM with basic inputs which includes IP Address and Port.

These fields would be referred for communicating or polling the details from the appliances via the other functions of SHM.

## Collect data periodically

The SHM client can request/poll for details from the appliances managed by SHM periodically or randomly to understand the current / live status of the polled/queried appliance. Based on the YANG specification and the exposed fields, the client would be returned with the appropriate values back from SHM server.

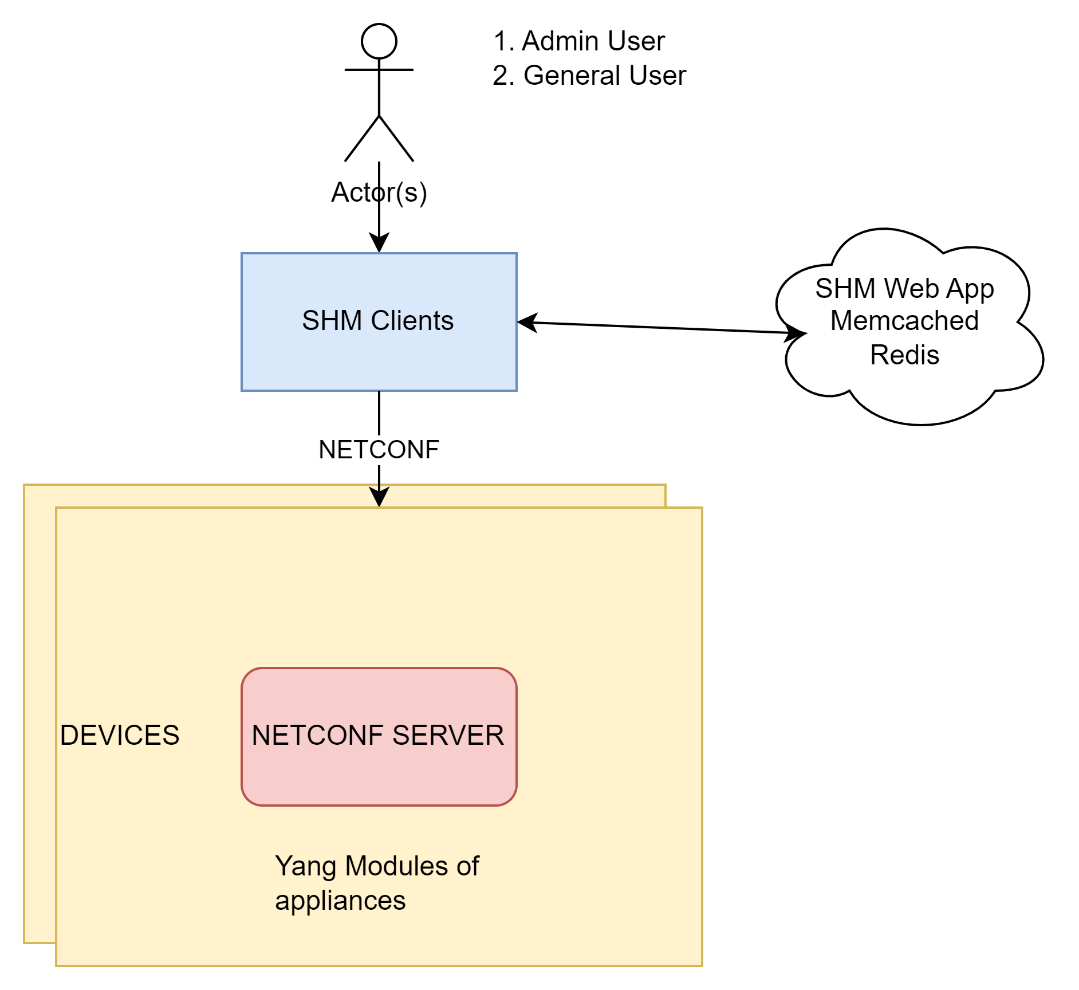
## Troubleshoot and control the devices

The SHM Server also offers facility to control the life cycle of appliance by sending out the action commands to the device. Example: start, stop, restart, dump-trace etc.

## Visualize the device details and provide some metrics

The SHM can come up with intuitive UI for showing the dashboard reflecting the details of the managed entities. The SHM can also extract / scrape for the metrics exposed by the appliances to be displayed on the UI. The choice of UI is yet to be explored, but the SHM shall be ready with the exposure of the metrics irrespective of the UI.

# Logical View



# Use Cases

|  |  |
| --- | --- |
| **Persona Map** | |
| Admin user | The person who Provides, Installs and Bills the SHM facility. |
| General user | The person who Consumes and controls the appliances using the SHM facility. |

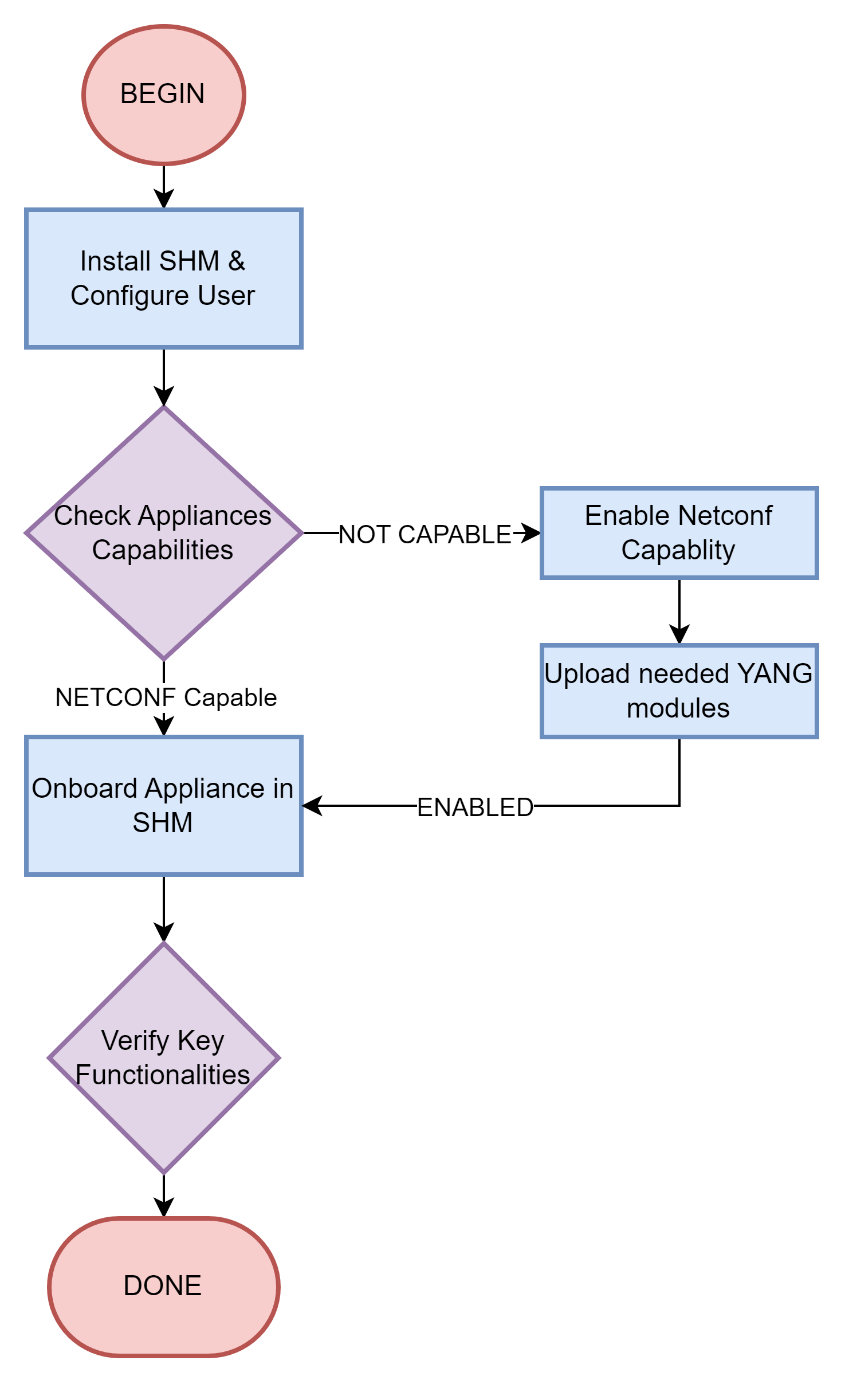
## As an Admin user, I should be able to configure/install the SHM at a premise.

## As a General User, I should be able to control my appliances using the SHM client.

## As a General User, I should be able to view & monitor my appliances using SHM Client.

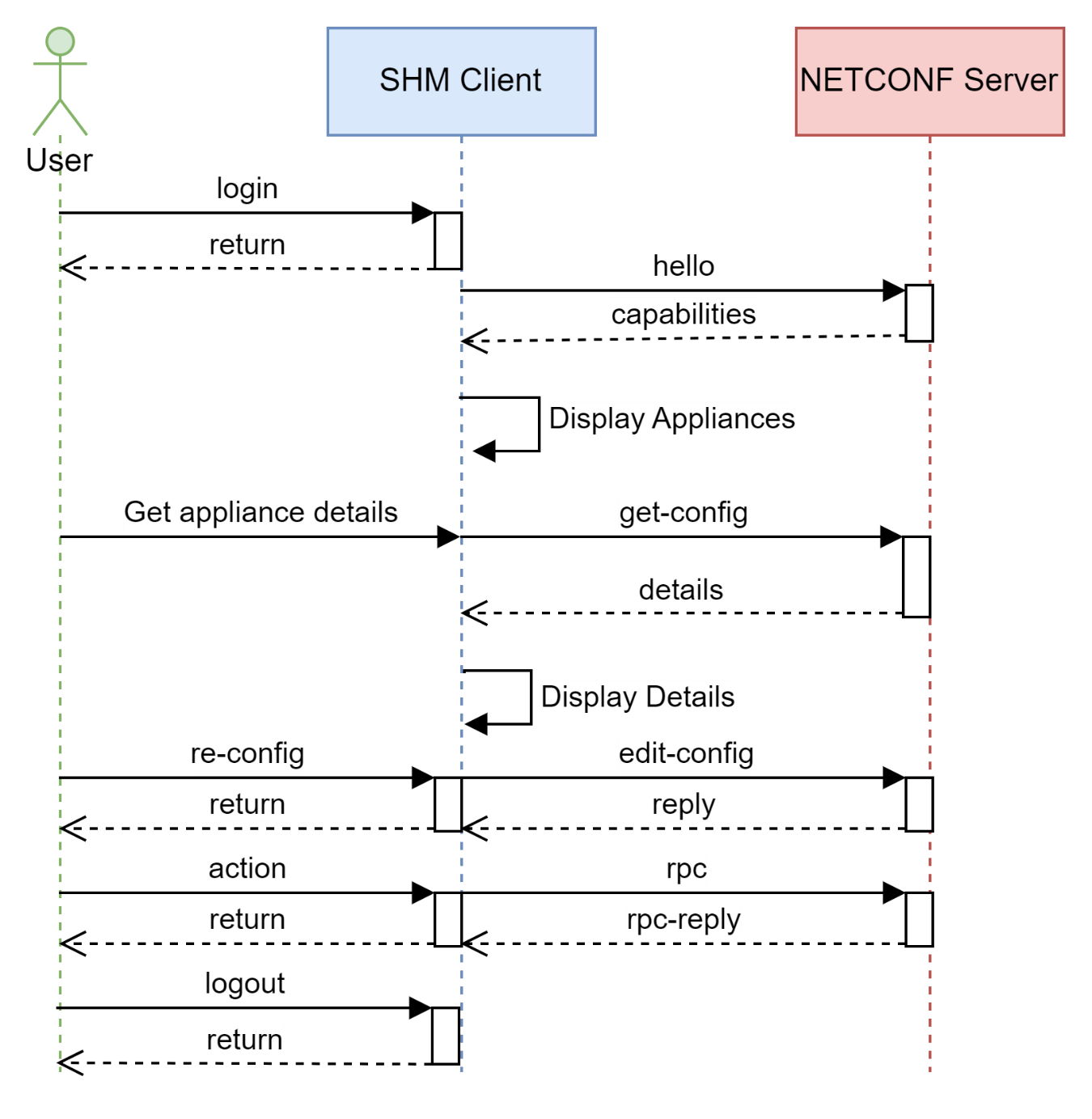
# Use Cases Detailed View

## As an Admin, I should be able to install and activate the SHM on premise.



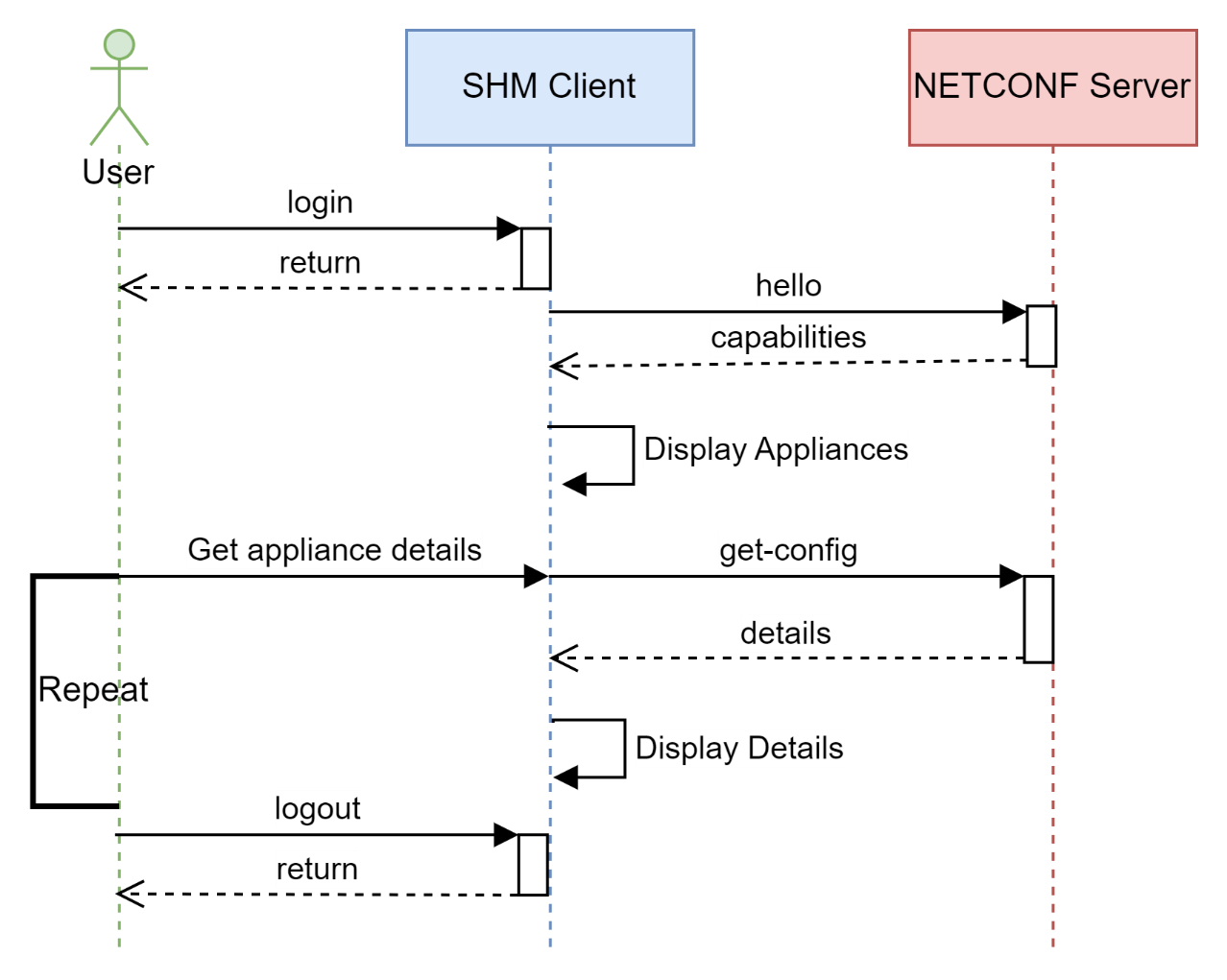
1. Install the SHM client / application on the smartphone or laptop.
2. Configure user and enable credentials and role mapping.
3. Find the list of appliances that needs to be managed by SHM.
4. Check for the NETCONF-YANG capabilities of the appliances.
5. Enable or adapt the appliance to support the NETCONF and upload the needed YANG modules to the appliances.
6. Add or onboard the appliance to the SHM client.
7. Verify key functionalities and handover SHM to the user.

## As a General user, I should be able to control the appliances.



1. Login into SHM Client using the credentials.
2. The SHM Client retrieves the capabilities and list of devices to show.
3. User requests for details about the appliances.
4. SHM Client retrieves the details using get-config.
5. Client displays the details to user.
6. User performs some re-configuration on the appliance.
7. Client performs the request using the edit-config and responds with return.
8. User perform some actions on appliance using client.
9. Client performs the action using the RPC calls and gets the reply.
10. Repeat from Step-3 or User Logs out.

## As a General user, I can view and monitor the appliances in my SHM client.



1. Login into SHM Client using the credentials.
2. The SHM Client retrieves the capabilities and list of devices to show.
3. User requests for details/metrics exposed by appliances.
4. SHM Client retrieves the details using get-config.
5. Client displays the details to user.
6. User can repeat steps 3 to 5 periodically.
7. User can logout of the SHM Client.

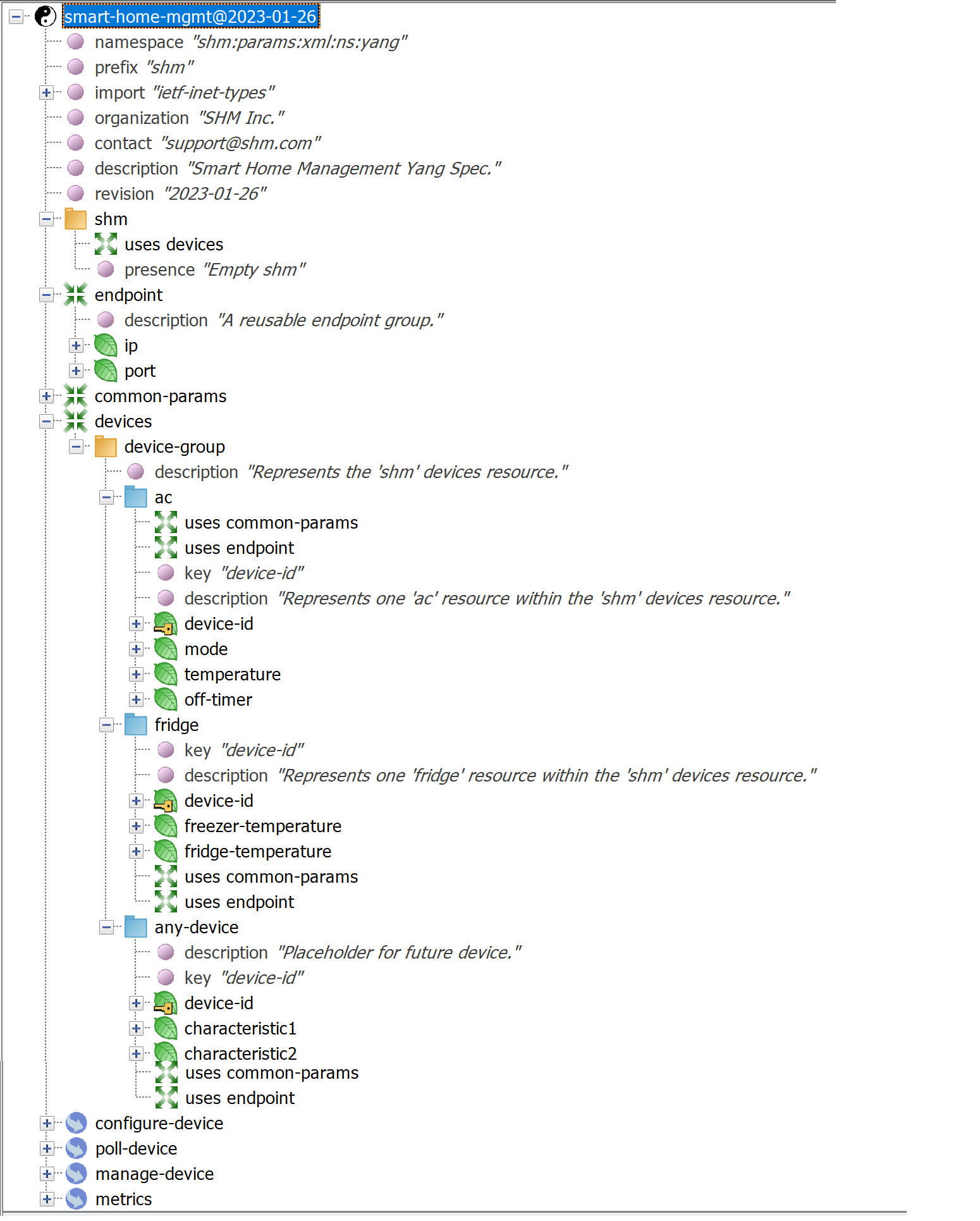
# 7. Information model

This section describes about the YANG Specification and various entities used in the data model. The data model is represented in UML Class diagram, YANG spec and also in tree view.

Below are few keywords and their relevant usage in the YANG specification file.

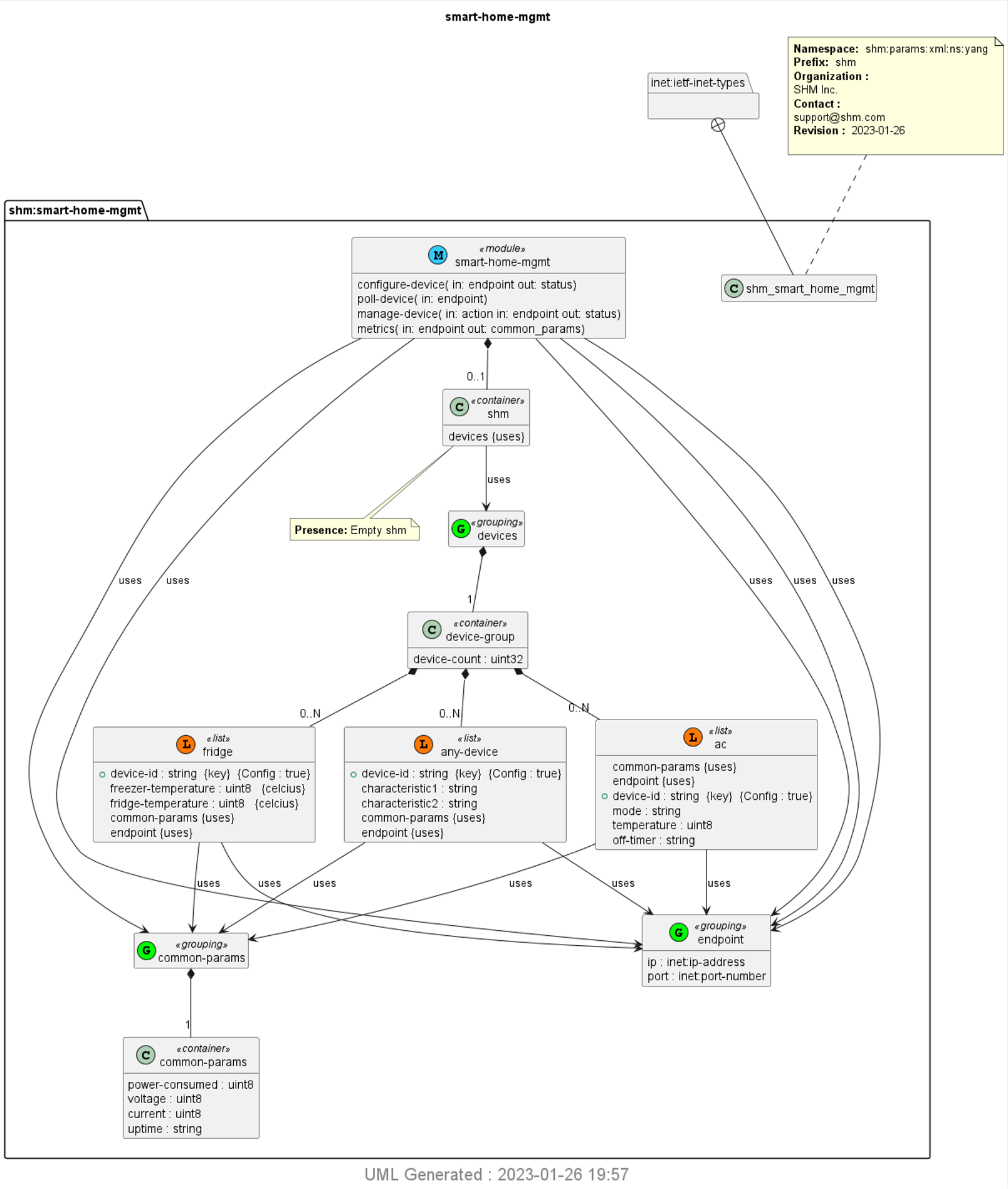
1. Module – Represents the base module (smart-home-mgmt in our case)
2. Grouping – To use some common grouped items (endpoint, common-params)
3. Containers – Represents the managed entities (Device-group in our case)
4. Lists – The logical grouping of related params for entities (ac, fridge, any-device)
5. Leaf – Represents the common and specific attributes of the managed entities.
6. RPC – Represents the actions that could be done on the managed entities.
7. Input & output – Represents the input and output supplied for RPC calls
8. Choice – Cases – Represents the conditional output based on entity.

**TREE View (Yang Modeler 3PP)**



|  |  |
| --- | --- |
| **YANG SPEC View (PYANG)**  module: smart-home-mgmt-new  +--rw shm!  +--rw device-group  +--rw ac\* [device-id]  | +--rw common-params  | | +--rw power-consumed? uint8  | | +--rw voltage? uint8  | | +--rw current? uint8  | | +--rw uptime? string  | +--rw ip? inet:ip-address  | +--rw port? inet:port-number  | +--rw device-id string  | +--rw mode? string  | +--rw temperature? uint8  | +--rw off-timer? string  +--rw fridge\* [device-id]  | +--rw device-id string  | +--rw freezer-temperature? uint8  | +--rw fridge-temperature? uint8  | +--rw common-params  | | +--rw power-consumed? uint8  | | +--rw voltage? uint8  | | +--rw current? uint8  | | +--rw uptime? string  | +--rw ip? inet:ip-address  | +--rw port? inet:port-number  +--rw any-device\* [device-id]  | +--rw device-id string  | +--rw characteristic1? string  | +--rw characteristic2? string  | +--rw common-params  | | +--rw power-consumed? uint8  | | +--rw voltage? uint8  | | +--rw current? uint8  | | +--rw uptime? string  | +--rw ip? inet:ip-address  | +--rw port? inet:port-number  +--rw device-count? uint32  rpcs:  +---x configure-device  | +---w input  | | +---w ip? inet:ip-address  | | +---w port? inet:port-number  | +--ro output  | +--ro status? string  +---x poll-device  | +---w input | | | +---w ip? inet:ip-address  | | +---w port? inet:port-number  | +--ro output  | +--ro (device)?  | +--:(ac)  | | +--ro temperature? uint8  | | +--ro mode? string  | | +--ro off-timer? string  | +--:(fridge)  | +--ro freezer-temperature? uint8  | +--ro fridge-temperature? uint8  +---x manage-device  | +---w input  | | +---w ip? inet:ip-address  | | +---w port? inet:port-number  | | +---w action? string  | +--ro output  | +--ro status? string  +---x metrics  +---w input  | +---w ip? inet:ip-address  | +---w port? inet:port-number  +--ro output  +--ro common-params  | +--ro power-consumed? uint8  | +--ro voltage? uint8  | +--ro current? uint8  | +--ro uptime? string  +--ro (device)?  +--:(ac)  | +--ro temperature? uint8  | +--ro mode? string  | +--ro off-timer? string  +--:(fridge)  +--ro freezer-temperature? string  +--ro fridge-temperature? string |

**UML Class Diagram View (Plantuml)**



YANG Spec file and below files could be located in the below Github repo.

<https://github.com/srivatzm/Smart-Home-Mgmt>

1. README.md – Actual Requirements received for this Smart home management.
2. SHM.drawio – Contains all the diagrams used in Use case view
3. Smart Home Management - System Design.docx – Design document.
4. smart-home-mgmt.png – Class diagram
5. smart-home-mgmt.uml – UML file
6. smart-home-mgmt.yang – YANG File.

# Security

The implementation of the NETCONF – YANG shall follow the security considerations laid out in the RFC4741 guidelines which can be located at [RFC 4741: NETCONF Configuration Protocol (rfc-editor.org)](https://www.rfc-editor.org/rfc/rfc4741#page-64).

The implementation of the LOGIN shall follow the appropriate IDM/IAM guidelines based on the 3PP selected for the same.

The SHM didn’t find any other venues of vulnerability with the current minimal requirements, but at the same time, the security should be hardened based on the future evolution as necessary.

The privacy details should be maintained as per the rules laid specific to the country where the SHM is implemented. GDPR in EU/EEA, PDP in India etc.

# Wireframe UI Mockups

|  |  |  |  |
| --- | --- | --- | --- |
| |  | | --- | | Smart Home Management  Login  Username :  Password :  Reset  Submit    Forgot credentials?  New User?  Sign Up | | |  | | --- | | Smart Home Management: Home  Welcome Tim  View Devices  Manage Devices  View Metrics  Configure Devices  Help?  Sign out | |
| |  | | --- | | Smart Home Management: View  Welcome Tim  Device Group-1  AC  Fridge  Back  Help?  Sign out | | |  | | --- | | Smart Home Management: Metrics  Welcome Tim  Device Group-1 > AC  IP:Port: 1.2.3.4:80  Mode: Normal  More..  Back  Help?  Sign out | |
| |  | | --- | | Smart Home Management: Action  Welcome Tim  Action: 1.2.3.4:80  Device-ID: AAAA  Command Console:  #status  UP  #restart  command received.  Back  Help?  Sign out | |  |

# Technology choice

|  |  |
| --- | --- |
| **Technology** | **Rationale** |
| UI | 1. App Based   1. Android  2. Ios  Easier usability for the end consumers.   1. Web Based 2. Angular JS – Widely used 3. React – Best for Single Page Application |
| Cache | Memcached  Simple key-value based distributed cache system used for storing some user preferences and also last polled device information at the client side. |
| Communication | NETCONF-YANG  Best preferred for install, manipulate and removal of network elements.  Works on top of RPC using XML based encoding.  Secured protocol when communicated over TLS and clients uses SSH. |
| Database | MongoDB is suitable for hierarchical data storage and is almost 100 times faster than Relational Database Management System  But we need an alternative if depth of an single element might exceed 100 levels.  Redis could be used considering we don’t have large amount of device management inside a premise. Redis could be deployed in cloud. |
| Cloud | GCP – Google cloud Platform is cost effective compared to AWS on billing methods.  GCP provides many out of the box analytics solutions which can help our SHM in giving many insights back to the consumer. |
| Web Application | Could be developed using Spring-Boot along with UI frameworks which could be reflected on the App or directly viewable at browser. |
| Plugins | We can use Quarkus in the place of Spring-boot for its fast boot time and hot-swap code deploy capabilities. Quarkus also provides many plug-ins for exposing the metrics, scraping the same by Prometheus and plotting them by Grafanna. |