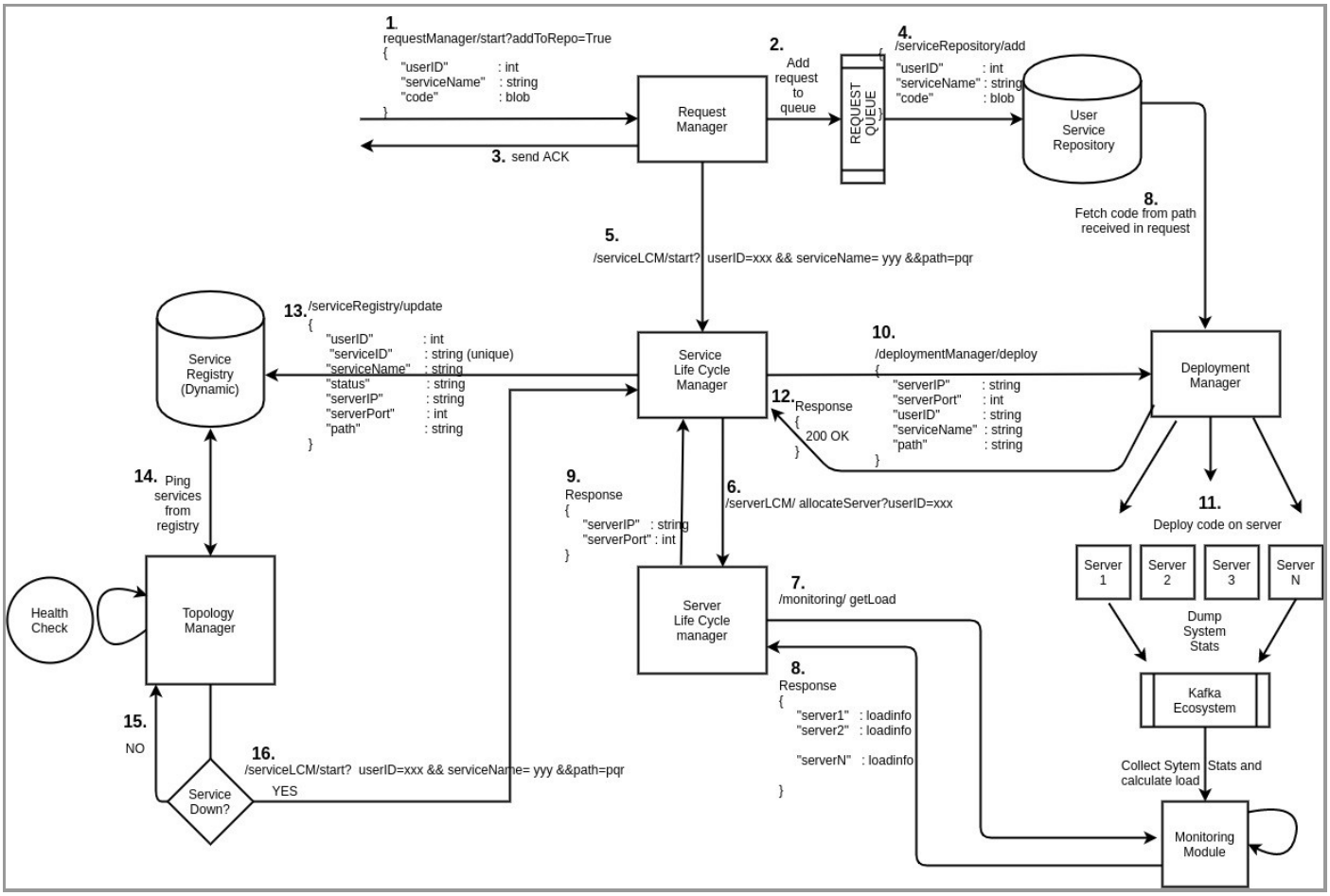
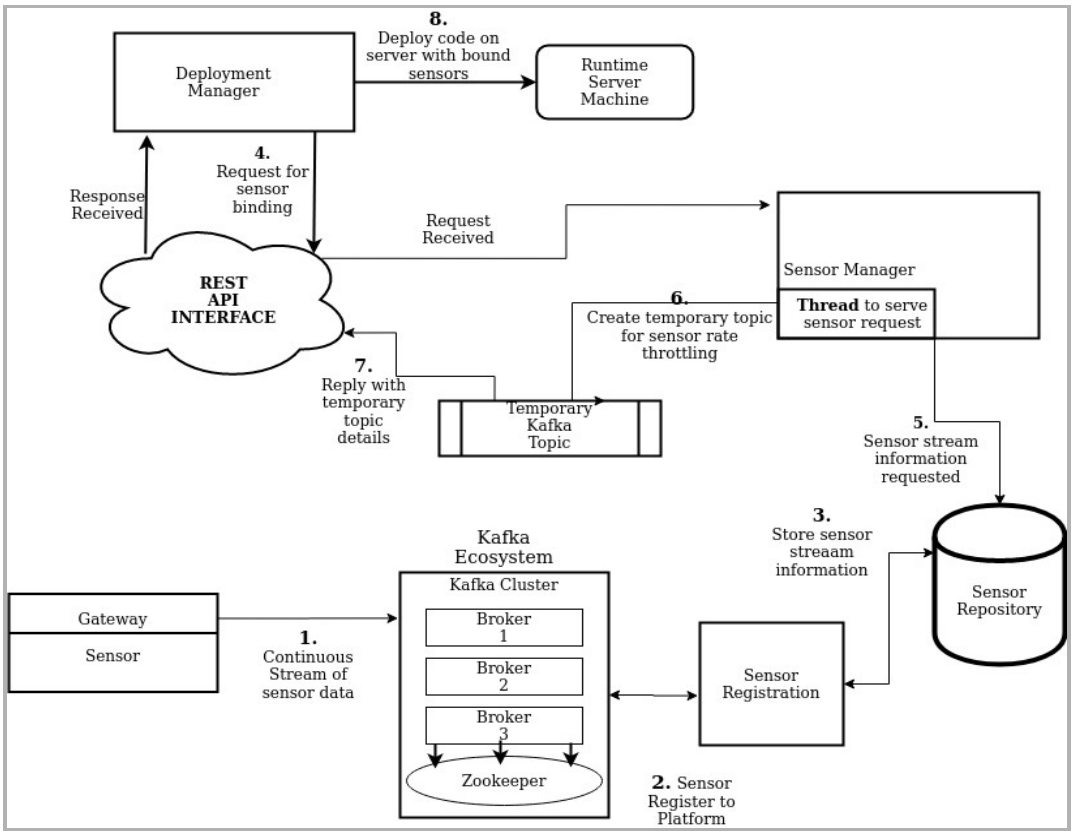
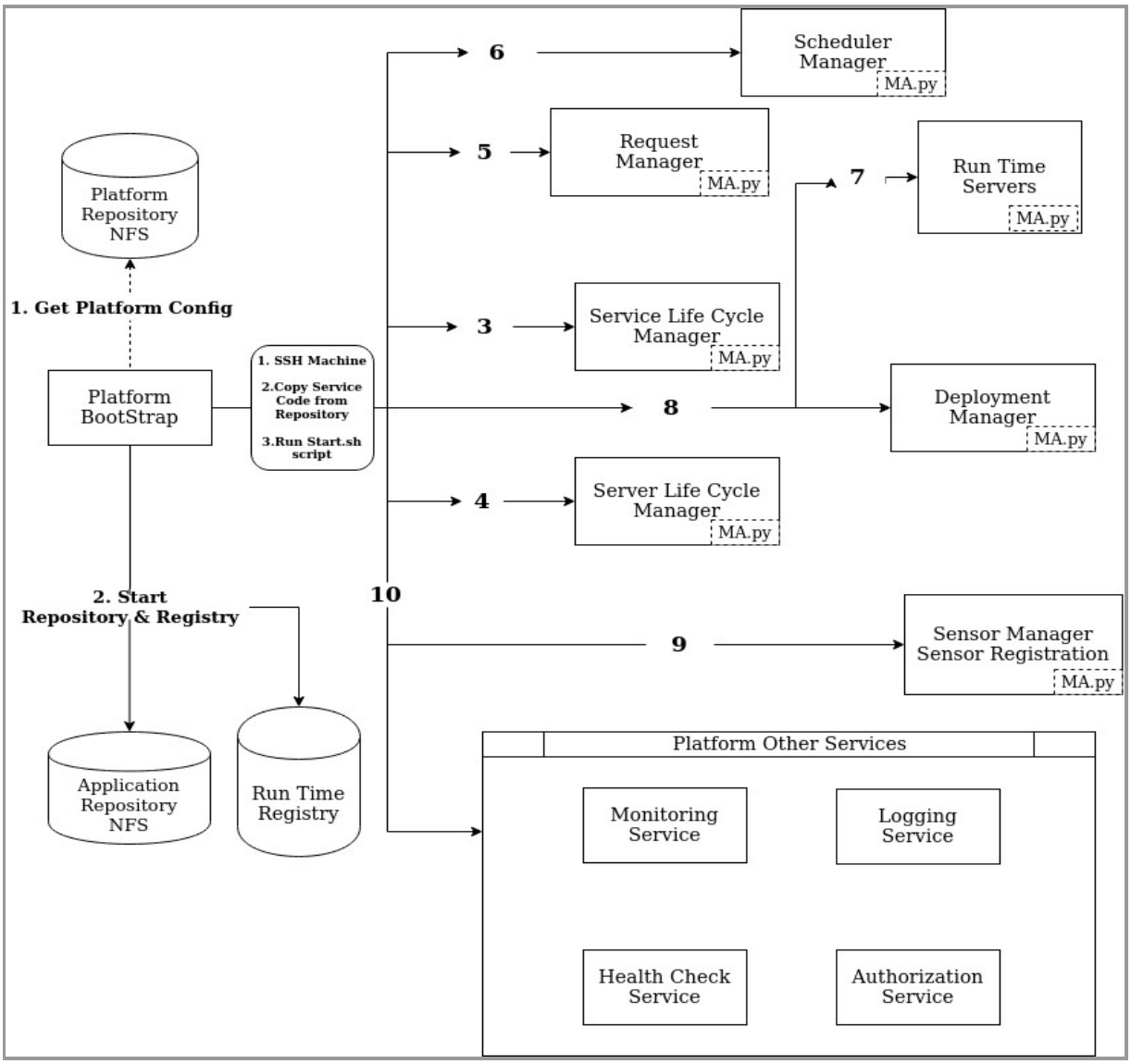
Internals of Application Server
Project Design
Group-1:

Communication Model

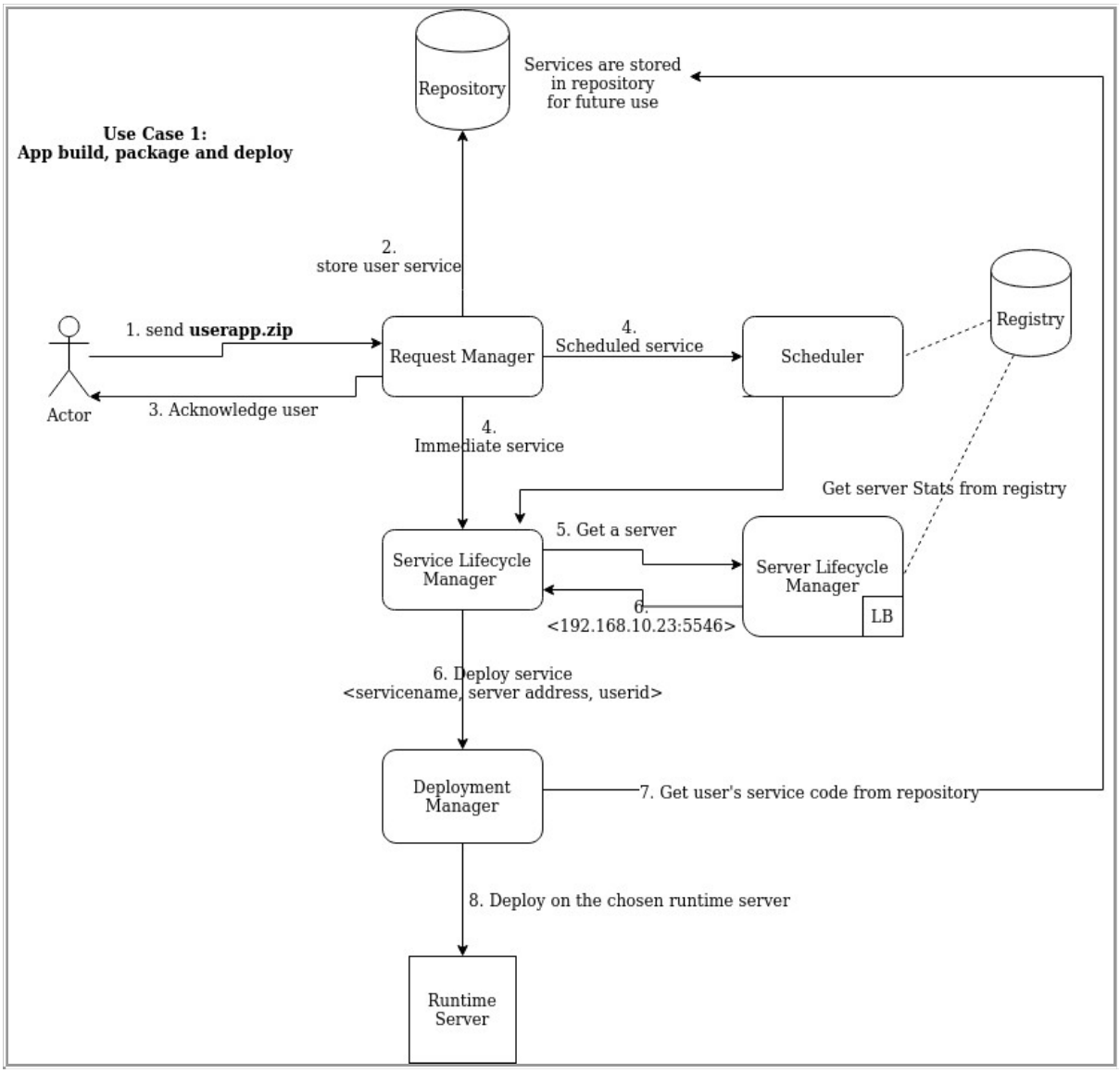


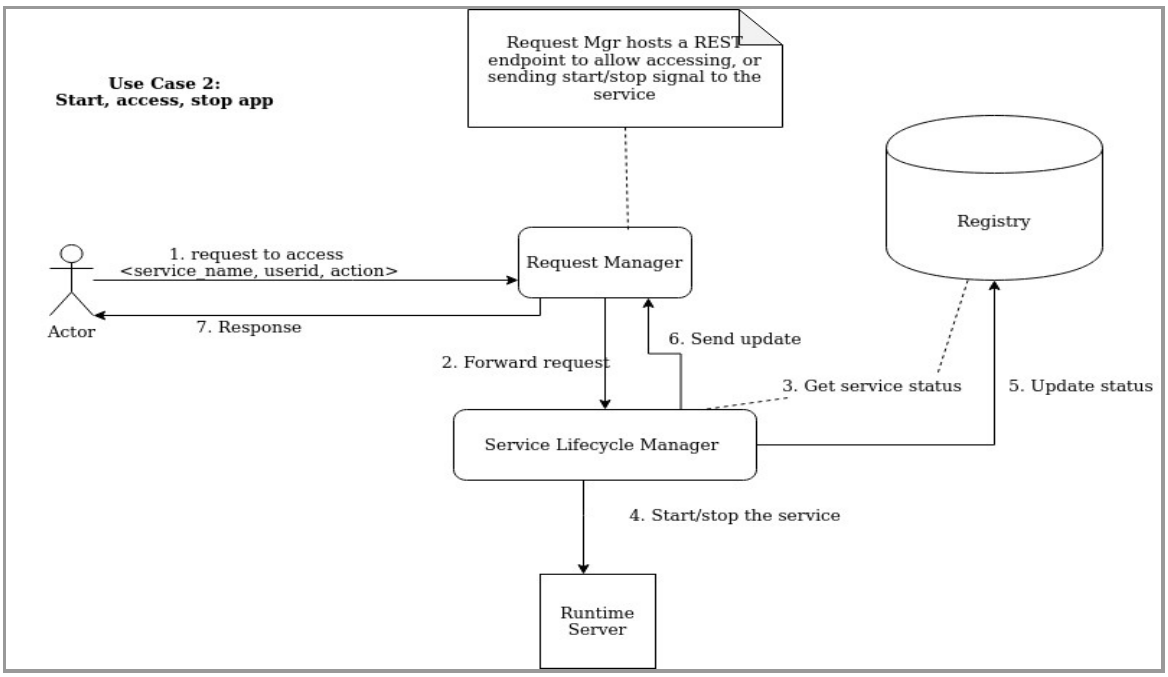


BootStrap Program



Application Model and Life Cycle:





Meta / Config Files

- User zip Directory Structure:

- Application1
  - Services
    - Service1
      - Code
        - Servicecode1.py
        - Servicecode2.py
      - Dependency
        - Ml_model_weights.pkl
        - File1.txt
  - Scheduling_info
    - Schedule1.xml
    - Schedule2.xml
  - Configuration_file.xml

Platform Code Directory:

- **inti.zip**
  - Config.yml
  - Machine Agent.py
  - Unused_IP.txt
  - Service Lifecycle/
  - Server LifeCycle/
  - Scheduler/
  - Deployment Server/
  - others/

Service Configuration File Format

```
<Services>
    <service>
        <name>service1</name>
        <filename>service1.py</filename>
        <priority>low</priority>

        <dependencies>
            <dependency>service2</dependency>
            <dependency>service3</dependency>
        </dependencies>
        <sensor_input>
            <input>
                <type>BY_ID</type>
                <id>12345</id>
            </input>
            <input>
                <type>BY_LAT_LON</type>
                <lat>1234.12</lat>
                <lon>24.213</lon>
                <floor>2</floor>
                <radius>12</radius>
                <count>ALL</count>
            </input>
        </sensor_input>
        <scheduling_info>
            <file_name>scheduling1.xml</file_name>
        </scheduling_info>
    </service>
</Services>
```

Platform Config File

```
BootStrap Config:

Machine IP

port

Host pswd

Services_to_run
    service Name: Server LCM
        root_dir : path of /serverLC
        docker_file: .dockerfile (container code for a particular service)
        startscript : .sh file (script to run docker file)
        params: json file

    service Name: Service LC M
        root_dir : path of /serverLC
        docker_file: .dockerfile
        startscript : .sh file
        params: json file
```

Scheduling File Format

```
<schedules>
    <schedule>
        <!-- Schedule 1 Info -->
    </schedule>
    <schedule>
        <!-- Schedule 2 Info -->
    </schedule>
</schedules>
```

Different Types Of schedules
SINGLE INSTANCE

```
<schedules>
    <schedule>
        <type>SINGLE_INSTANCE</type>
        <start_date>12-12-2020</start_date>
        <time>
            <start>10:00</start>
            <end>12:00</end>
        </time>
    </schedule>
</schedules>
```

DayWise

```
<schedule>
    <type>DAYWISE</type>
    <day>MONDAY</day>
    <time>
        <start>10:00</start>
        <end>12:00</end>
    </time>
</schedule>
```

Periodic

```
<schedule>
    <type>PERIODIC</type>
    <start_date>12-12-2020</start_date>
    <period>10</period>
    <time>
        <start>10:00</start>
        <end>12:00</end>
    </time>
</schedule>
```

Sensor Registration

```
<sensors>
    <sensor>
        <name>BTF-14</name>
        <gateway>ip:port</gateway>
        <sensor_id>SI12</sensor_id>
        <data_type>VECTOR</data_type>
        <size> 20 </size>
        <address>
            <area> iiit </area>
            <building_name>nilgiri</building_name>
            <room_no>D12</room_no>
        </address>
        <geo_location>
            <lat>1234.242</lat>
            <lon>854.21</lon>
            <floor_no>4</floor_no>
        </geo_location>
        <type>INPUT_OUTPUT</type>
    </sensor>
</sensors>
```

Deployment Manager To Sensor Manager

Format:

```
<sensors>
        <sensor>
            <!-- sensor 1 information -->
        </sensor>

        <sensor>
            <!-- sensor2  information -->
        </sensor>
</sensors>
```

BY ID:

```
    <sensor>
        <type>BY_ID</type>
        <id>12345</id>
        <rate>100</rate>

    </sensor>
```

BY_ADDRESS

```
<sensor>
    <type>BY_ADDRESS</type>
    <area>iiit</area>
    <building_name>nilgiri</id>
    <room_no> D12 </room_no>
    <name>TMP36</name>
    <count>ALL</count>
    <rate>100</rate>
```

BY Lat/Log:

```
   <sensor>
       <type>BY_LAT_LON</type>
       <lat>1234.12</lat>
       <lon>24.213</lon>
       <floor>2</floor>
       <radius>12</radius>
       <name>TMP36</name>
       <rate>100</rate>
       <count>ALL</count>
   </sensor>
```

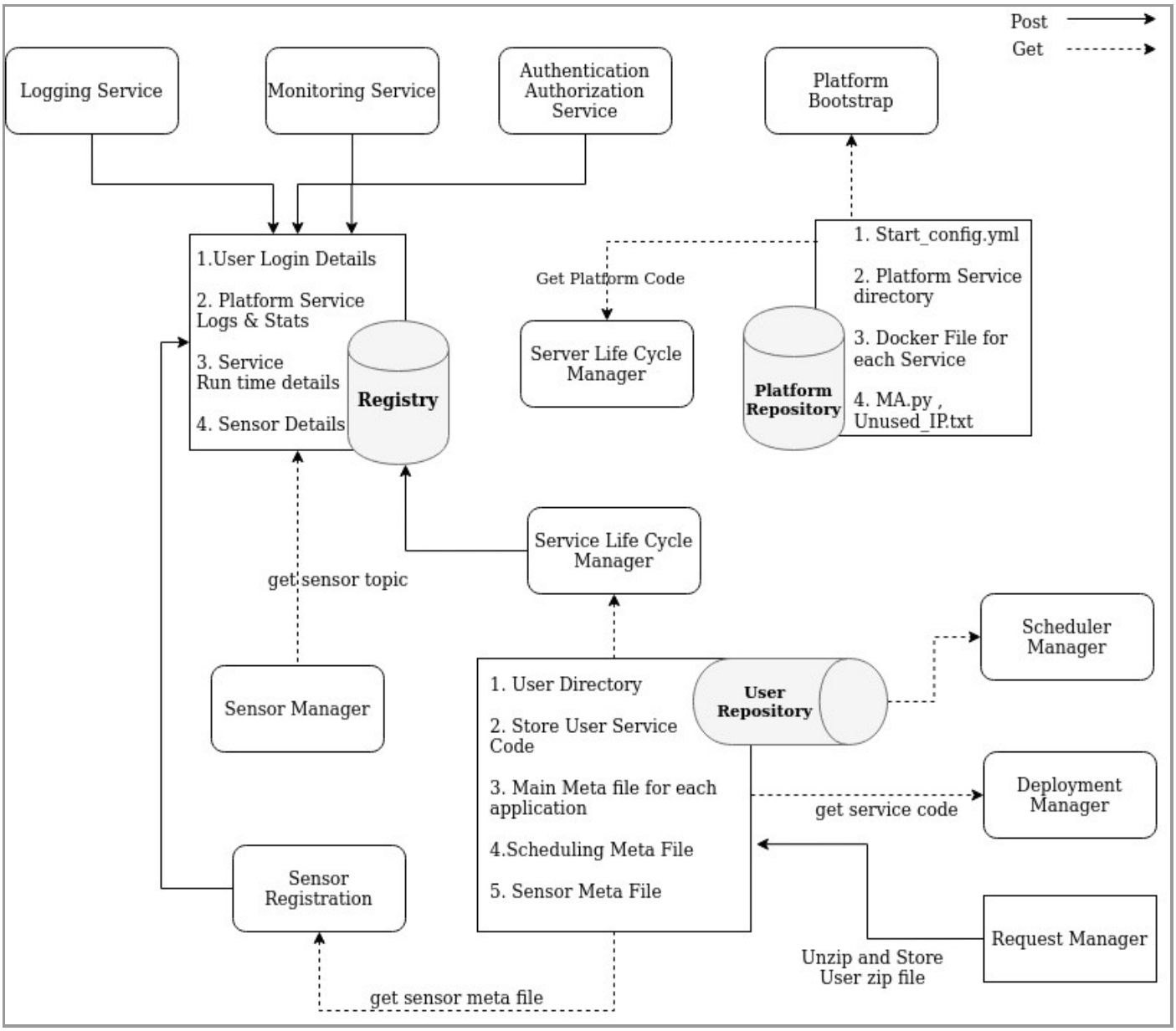
Service Life Cycle to Deployment Manager

Start a user service

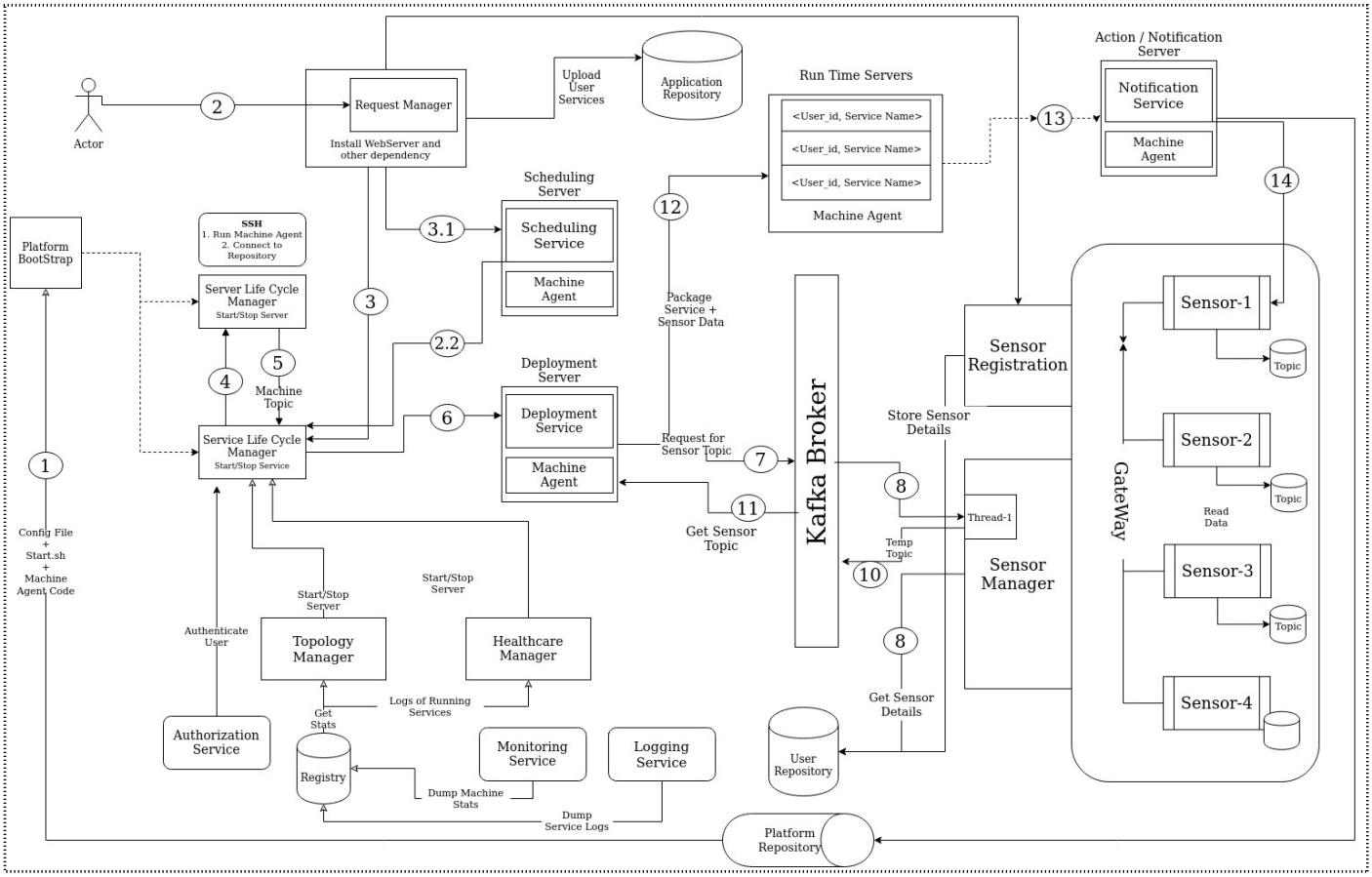
```
<service>
    <action>start</action>
    <user_id>group_1_user</user_id>
    <application>applicationname</application>
    <service_name>service1</service_name>
</service>
```

```
<service>
    <action>stop</action>
    <service_instance_id>axc12cas</service_instance_id>
</service>
```

Repositry and Registry:



Big Picture


Actor
2
Request Manager
Install WebServer and other dependency
Upload User Services
Application Repository
Run Time Servers
<User_id, Service Name>
<User_id, Service Name>
<User_id, Service Name>
Machine Agent
13
Action / Notification Server
Notification Service
Machine Agent
14
Scheduling Server
Scheduling Service
Machine Agent
3.1
12
Platform BootStrap
SSH
1. Run Machine Agent 2. Connect to Repository
Server Life Cycle Manager
Start/Stop Server
3
2.2
Package Service + Sensor Data
Sensor Registration
Sensor-1
Topic
Sensor-2
Topic
Sensor-3
Topic
Sensor-4
GateWay
Read Data
4
5
Machine Topic
Deployment Server
Deployment Service
Machine Agent
6
Service Life Cycle Manager
Start/Stop Service
1
Request for Sensor Topic
7
Kafka Broker
Store Sensor Details
8
Thread-1
Sensor Manager
Temp Topic
10
11
Get Sensor Topic
Config File + Start.sh + Machine Agent Code
Start/Stop Server
Start/Stop Server
Authenticate User
Topology Manager
Healthcare Manager
Logs of Running Services
Get Stats
Authorization Service
Registry
Monitoring Service
Logging Service
Dump Machine Stats
Dump Service Logs
User Repository
Get Sensor Details
Platform Repository