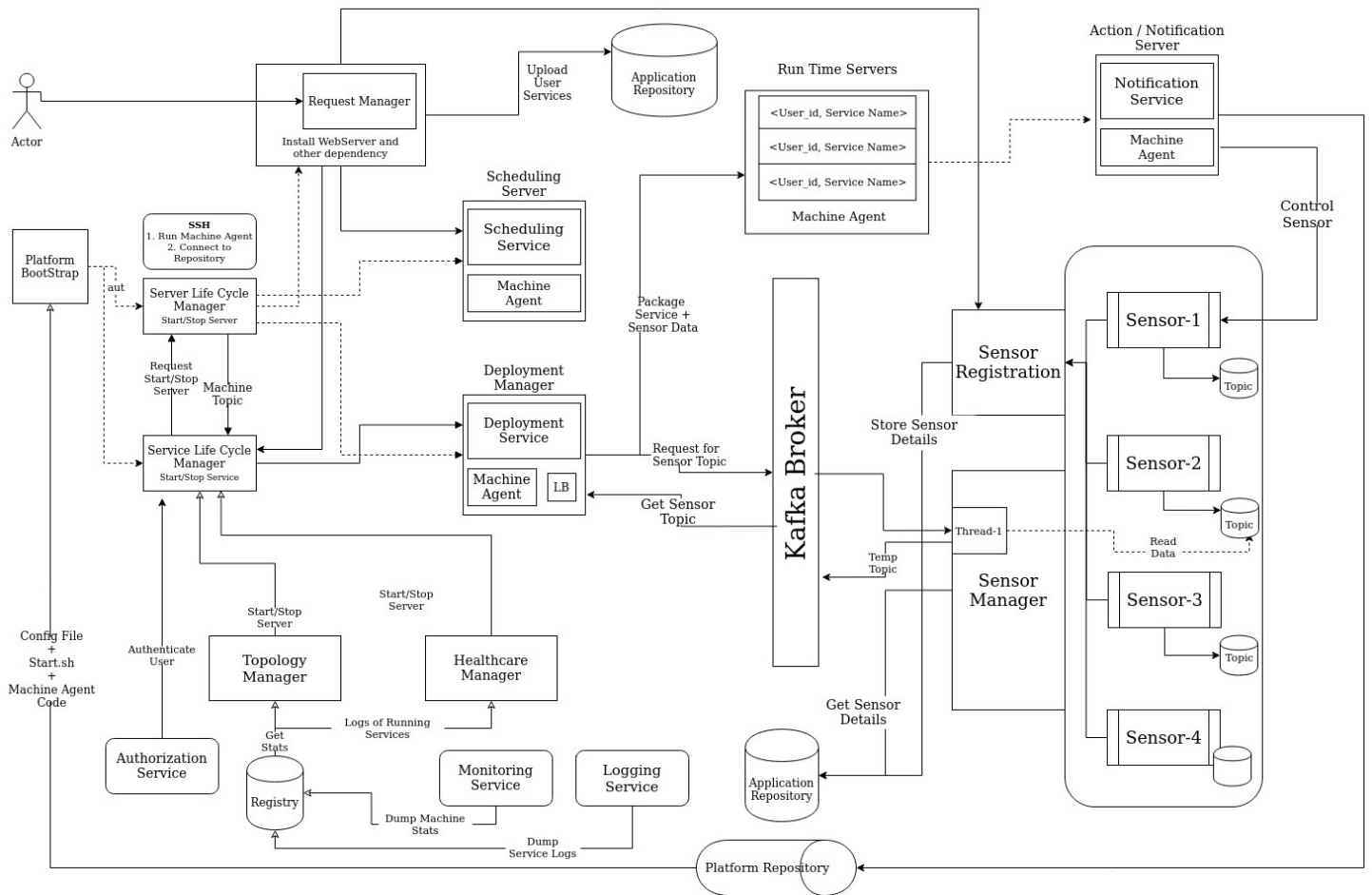


Internals of Application Servers

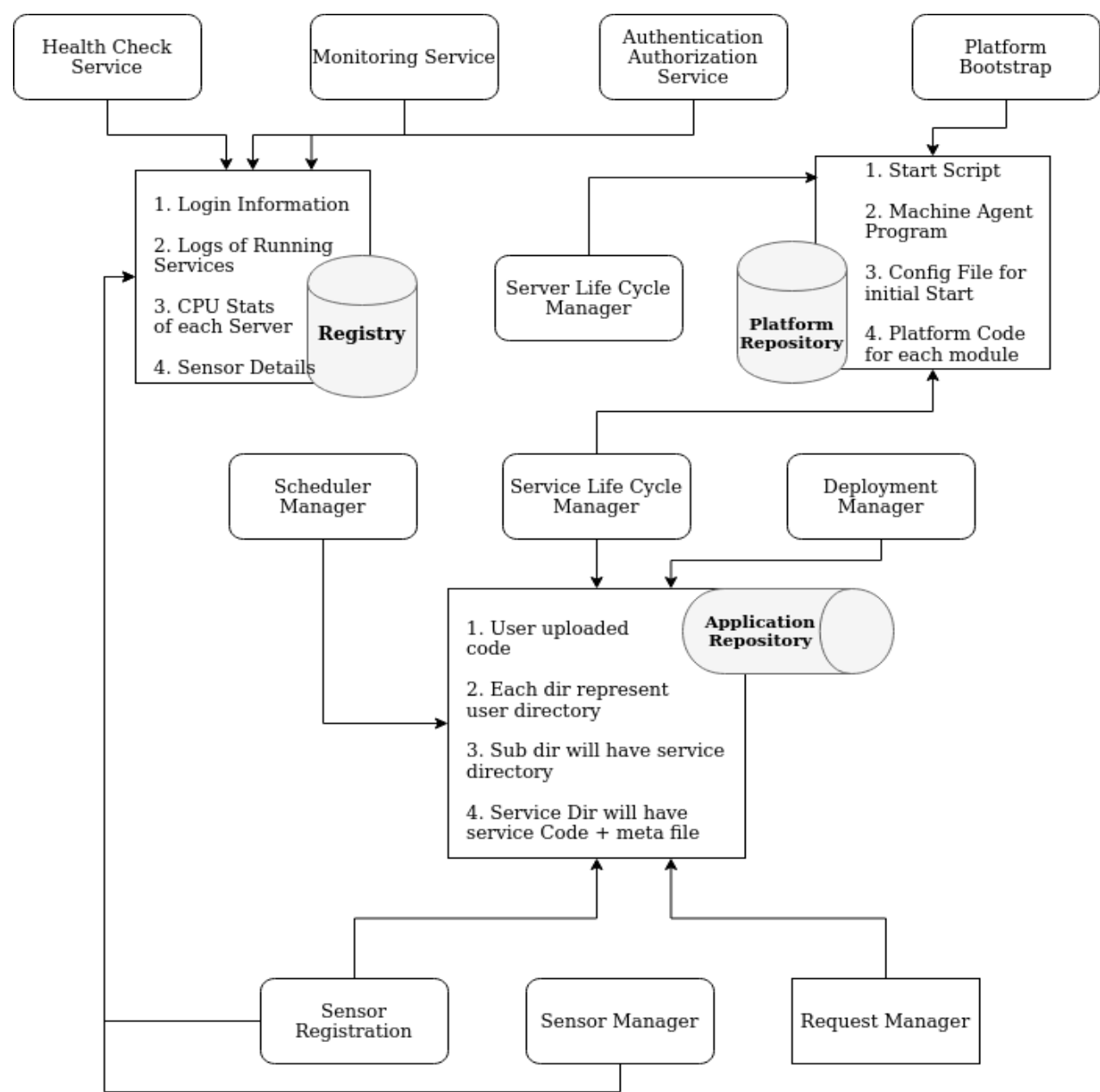
Project Design

Group-1

Big Picture:

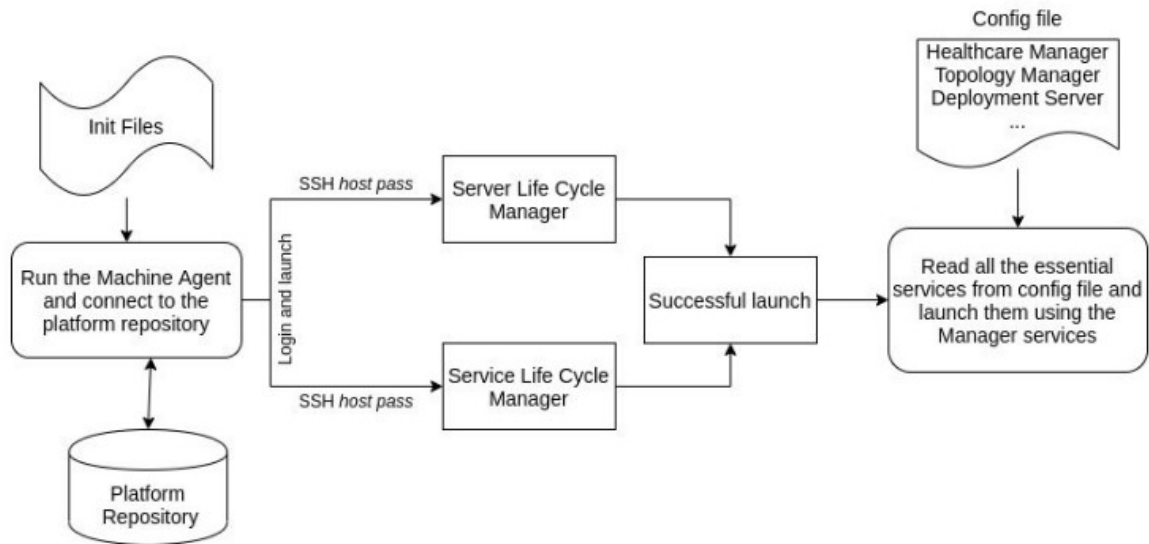


Registry / Repository:



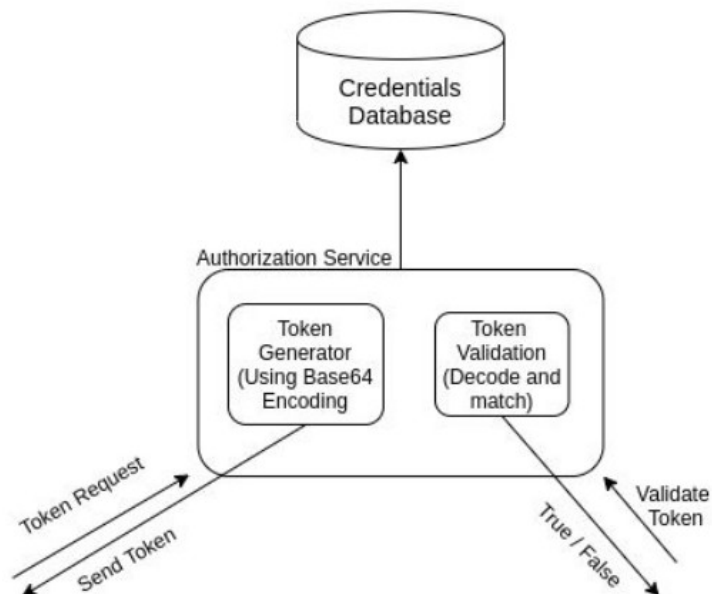
1. Platform Bootstrap

- The bootstrap manager performs all the initialization work. It is the first service launched.
- Bootstrap Manager first runs the machine agent and then connects with the repository.
- Using SSH, it runs the Service Lifecycle Manager and Server Lifecycle Manager services.
- All other required services for the platform to work are stored in the config file as <servicename, params>.
- These services are then launched one by one by the Bootstrap Manager.



2. Authorization Service

- There are 2 tasks for Authorization service.
 1. Generate Token
 2. Validate Token

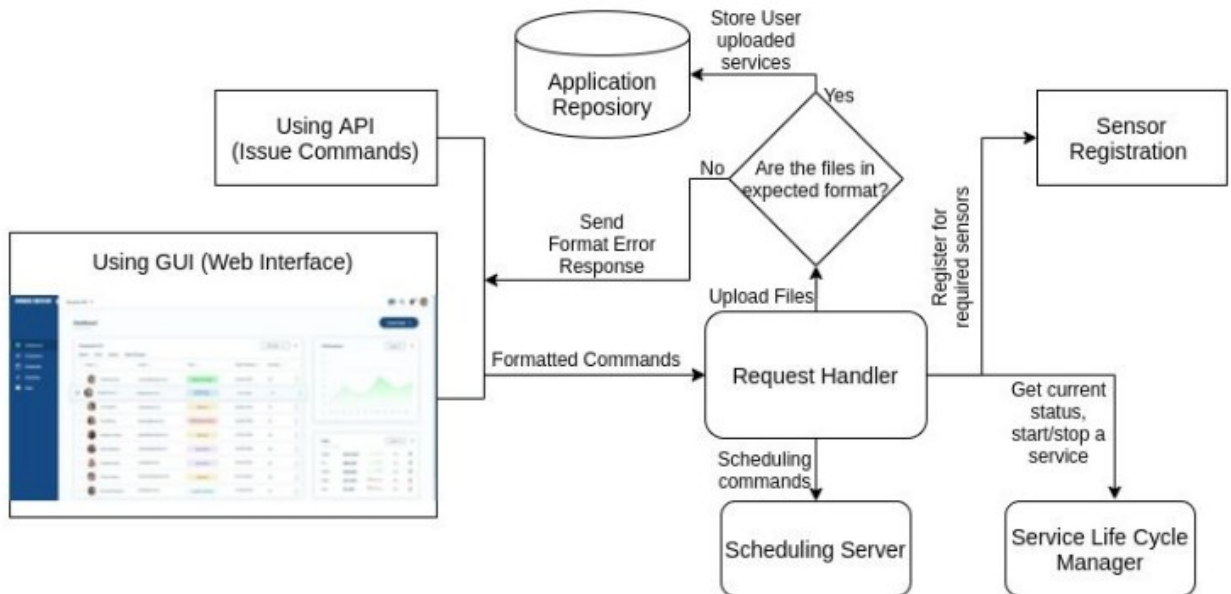


4. Request Manager

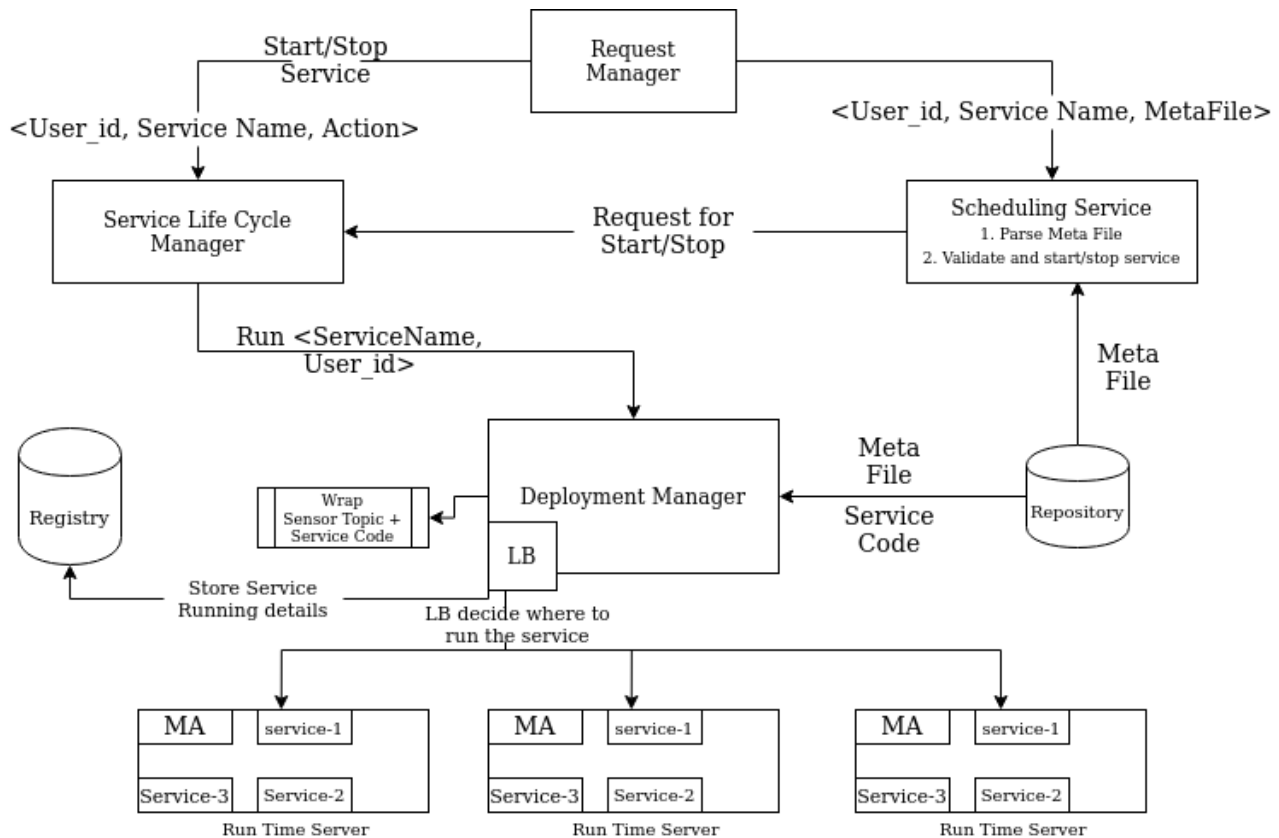
- This service accepts user requests, processes it and returns a response to the query.
- The request handler runs on Apache server, expects input commands in a particular format and generates response. Hence, it can be operated using both command-line/GUI as long as they comply with the format.

Commands:

- getFormatForUpload ◦ uploadFile ◦ listServices ◦ startService ◦ stopService ◦ scheduleService
- getInfoAboutService



Scheduler and Deployment Manager:

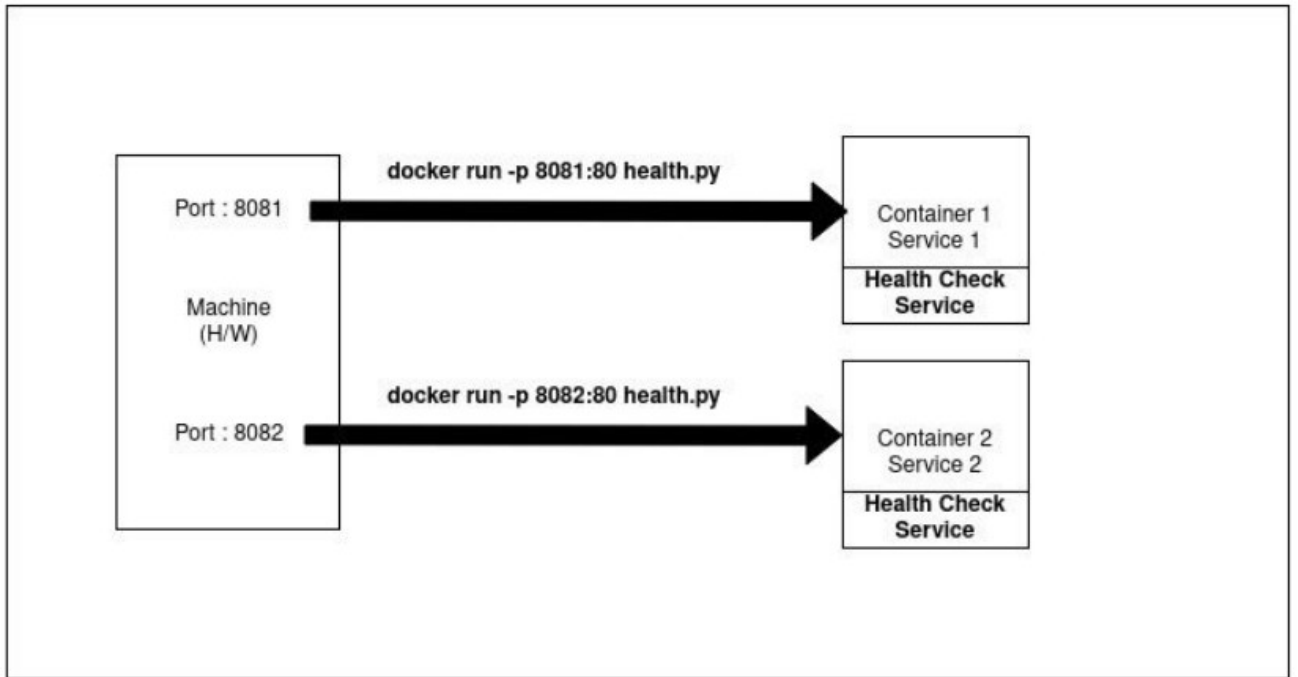


Health Check Service

1. This service is responsible for sending the health status of the service along which it is deployed.
2. It is invoked by a GET request and sends a success code (200 OK) if the service is up and running.
3. In case the service is down, this endpoint will be unreachable resulting in a response code starting with 4XX or 5XX.
4. The topology manager continuously polls this service in order to get the system diagnostic and takes appropriate actions based on the service's response as summarized in the table below.

Requirements to deploy health check:

1. Every container runs a **health.py** script which contains an endpoint **http://ip:port/health**
ip = hostip and port = hostToContainerForwardedPort

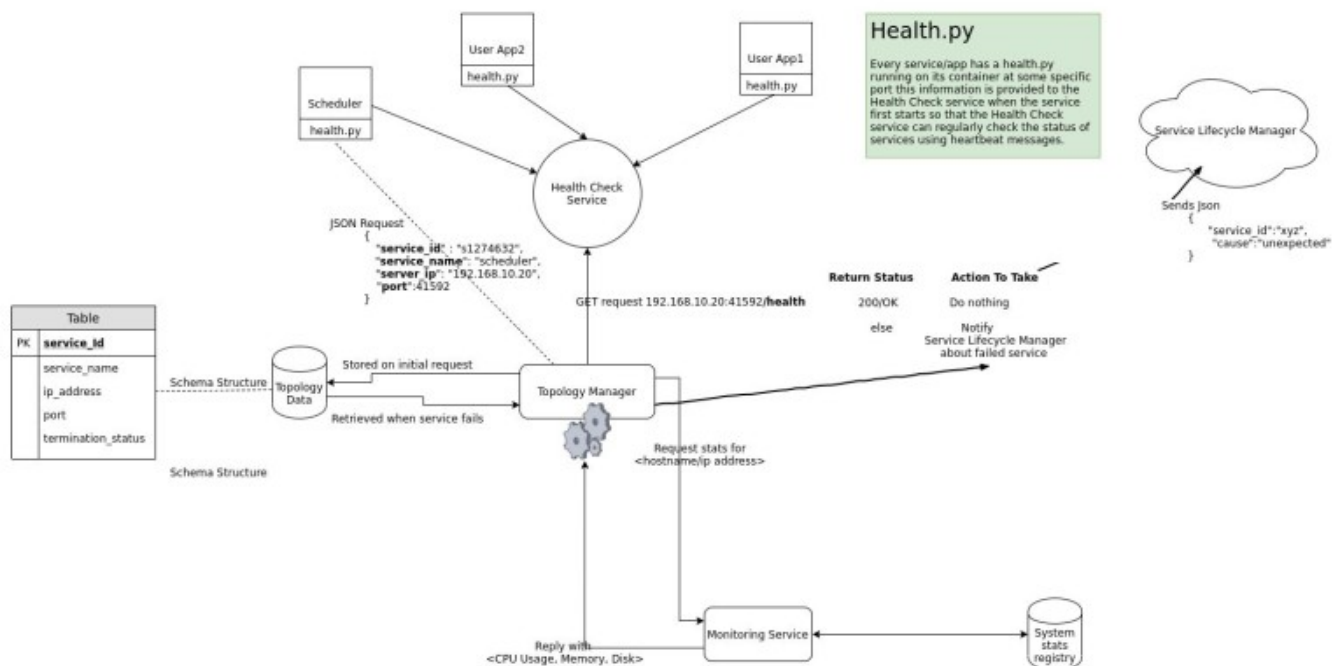


An example of how Health Check Service is deployed within each container with port forwarding
(COMPLETE DATA FLOW DIAGRAM BELOW)

Invoking Health Check Service (Data Flow between Modules)

Response	Action taken by Topology Manager
200/OK True	Do nothing
Other	Invoke <u>Service Lifecycle Manager</u> to re-deploy the service.

Topology Manager



1. Topology Manager has information about the entire platform's network topology.
2. Each service(user/platform service) that is deployed on the platform registers itself with the topology manager.
3. After which, Topology Manager gets information about the exact server and port at which the service(user/platform) is deployed. This information is later used by the manager to uniquely identify the service if it needs to be re-deployed.
4. Topology Manager polls the network graph to obtain system diagnostics about each module/service which was previously registered with it by hitting the Health Check service endpoint, as described in the above section.
5. If the health check returns a success code (200 OK true) this means that the service is up and running and nothing needs to be done.
6. In case of an alternate response code, Topology Manager invokes Service Life Cycle Manager to re-deploy the affected service.
7. In addition to this, the topology Manager uses statistics from the Monitoring Module to decide which servers are overloaded and sends a message to server or service lifecycle manager to start instances of servers which exceed high mark.

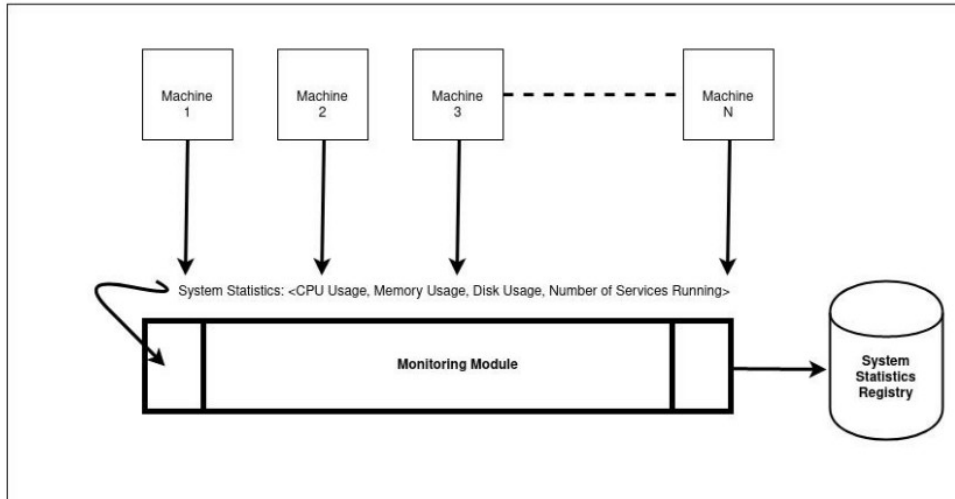
Input data format for service registration:

```
{
  "serviceId" : SomeUniqueIdentifier
  "serviceName" : xyz
  "serverIp" : 192.198.1.1
  "Port": hostToContainerForwardedPort
  "TerminationStatus": safe/unsafe or expected/unexpected
}
```

Actions:

1. Invoking Service LC Man with appropriate data format

Monitoring Service



1. The Monitoring Service is responsible for collecting system statistics from all the machines across the platform.
2. This module then converts the data into a pre-fined form and writes it into a registry.
3. Each entry in the registry is of the form : <CPU Usage, Memory Usage, Disk Usage,Number of Services>.
4. This data is then read by the Topology Manager to decide which servers are overloaded and exceeding high mark so that their instances can be deployed again.

Communication Model:

