# Purpose

Real-time data analytics to detect abnormal heart rate and send notification to registered contact.

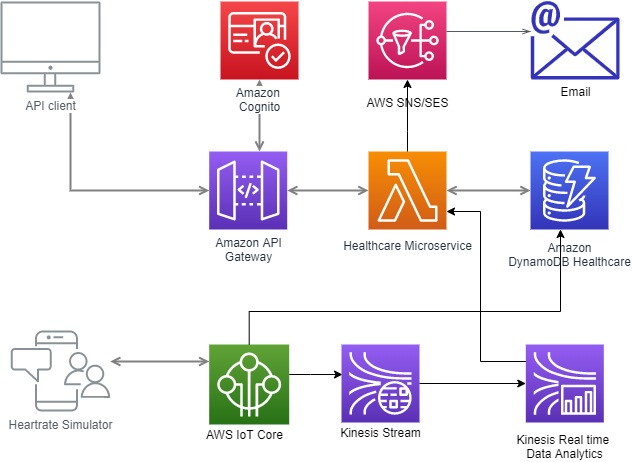
# System Description

Healthcare API gateway allow to register patient, provider and heartrate notification configuration data.

Heartbeat simulator sends heartbeat data to AWS IoT using mqtts which further ingest by real-time analytics engine and Heartbeat microservice to send notification to registered contact.

Healthcare professional can view patient’s last 10 minutes heartrate data using API gateway.

# High level architecture diagram



# Data Model and API Gateway

Healthcare professional register patient heartrate notification configuration info into system to send out auto notification alert.



# Patient registration

API URL <http://healthcare/patients>

METHOD: HTTP POST

|  |  |
| --- | --- |
| Patient info | |
| id | string |
| *providerId* | *String* |
| dateOfBirth | date |
| firstname | string |
| lastname | string |
| email | string |
| phoneNumber | string |
| address | string |
| admitdate | date |
| dischargedate | date |

# Last 10 minutes patients heartrate

API URL [http://healthcare/patients/{patientId}/heartrate](http://healthcare/patients/%7bpatientId%7d/heartrate)

METHOD: HTTP GET

|  |  |
| --- | --- |
| Heartrate response :- heartrate data | |
| timestamp | timestamp |
| *value* | *int* |
| *deviceId* |  |

# Provider API

API URL http://healthcare/providers

METHOD: HTTP POST

|  |  |
| --- | --- |
| Register Provider | |
| *id* | *String* |
| firstname | string |
| lastname | string |
| email | string |
| mobile | string |
| address | string |
| specialization | string |
| active | boolean |

# Real-time heartrate analytics

Kinesis stream receive heartrate data from AWS IoT core and Kinesis data stream used to execute heartrate rules on **SOURCE\_SQL\_STREAM\_001 to generate notification alert.**

**For any abnormal heartrate heartrate microservice lambda retrieve notification info from database to send out alert notification.**

**DynamoDb Table**

|  |  |
| --- | --- |
| **Device** | |
| Field | Details |
| id | Device ID “UUID” string**, partition key** |
| Type | Sensor type as sort key |
| deviceName | Device name, string |
| type | SensorType e.i Heart\_RATE |
| ipAddress | String; |
| port | integer |
| protocal | String, supported protocol; |

|  |  |
| --- | --- |
| **Patient** | |
| Field | Details |
| id | **partition key, Patient’s Base64 encoded email ID** |
| provider\_id | provider id **sort key** string , **Provider’s Base64 encoded email ID** |
| firstname | String |
| lastname | String |
| mobileNumber | String |
| address | String |

|  |  |
| --- | --- |
| **Patinet\_Device mapping** | |
| Field | Details |
| patientId | **partition key, Patient’s Base64 encoded email ID** |
| deviceId | Device id as sort key |
| sensonType | SensorType e.i Heart\_RATE |
| status | ACTIVE/INACTIVE |

|  |  |
| --- | --- |
| **provider** | |
| Field | Details |
| id | **partition key, Provider’s Base64 encoded email ID** |
| **mobile\_number** | **sort key**, provider mobile number. |
| firstname | String |
| lastname | String |
| specilization | String; |
| Mobile\_number | String |
| address | String |
| status | ACTIVE, INCATIVE |

|  |  |
| --- | --- |
| **Heartrate** | |
| Field | Details |
| deviceId | **partition key,** deviceId |
| timestamp | Sort key |
| payload | Heartrate data from heartrate simulator, contains heartrate and patientId |