Grp-4

VIVEKANAND EDUCATION SOCIETY'S INSTITUTE OF TECHNOLOGY

CHEMBUR, MUMBAI 400074

Case-Study Report on IoT Application

Forest Fire Detection System

SUBMITTED BY

NAME	ROLL NO
Nishtha Joshi	25
Anket Kadam	26
Anushka Karhadkar	27
Payoshni Khekale	28
Kaushik Kotian	29
Tejaswi Koul	30
Priyank Kulkarni	31
Aniket Mahajan	32

OF

B.E (D20-B)

DEPARTMENT OF INFORMATION TECHNOLOGY

UNDER THE GUIDANCE OF

Dr. Abhay Kshirsagar

INDEX

1: PURPOSE AND REQUIREMENT SPECIFICATION (Nishtha)

A forest fire detection system would be put in place to ensure early detection of wildfire with rapid responses and reduced damage from such fire. Such systems would constantly monitor large areas of forests using sensors, satellite images, drones, and real-time data analysis. Detection of fire at an early stage would facilitate this system to protect the environment, human life, wildlife, and nearby communes.

Requirements:

1. **Real-time monitoring:** The system should be able to provide real-time monitoring to detect fires as early as possible.

2**. Wide area coverage:** from large forested regions using several sensor types or satellite data.

3. **Accuracy:** The detection should be with high accuracy and without a lot of false alarms, which might be generated due to weather conditions like fog or dust.

4. Data Integration: The system should integrate data from various sources, including temperature sensors, wind patterns, and satellite imagery.

Challenges and Problems in Existing Systems

1. Delayed Detection: Most of the existing systems detect fires only when they are large enough to be visible, either because of smoke and heat signatures or flames. Delayed detection allows fires to grow, causing much damage before it is fought.

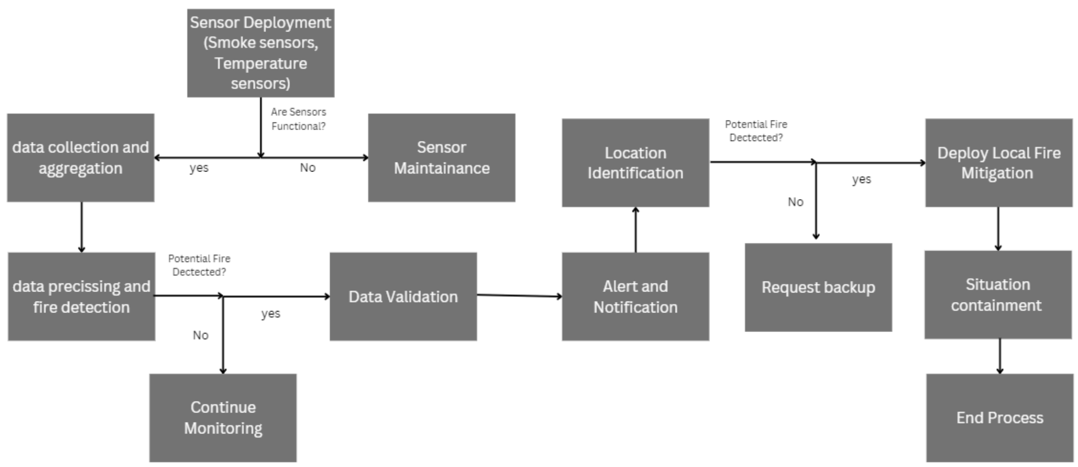
2. High False Alarm Rates: Most of the conventional satellite imagery and camera-based detection systems have a high rate of false alarms, which usually are from non-fire related phenomena such as dust, fog, or heat sources that are not on fire, wasting other resources.

3. Partial Coverage: Remote and vast areas are not covered by current systems due to infrastructural barriers, the high cost of installation, and technological limitations, especially in areas with low connectivity or access.

2: <u>PROCESS SPECIFICATION</u> (Anket Kadam)

Process Specification for Forest Fire Detection System:

1. **Sensor Deployment and Calibration:**
   - The system begins with the deployment of multiple sensors in the forest area. These sensors include temperature, smoke, and humidity sensors, which are strategically placed to cover the entire region.
   - Calibration of these sensors ensures accuracy in the detection of fire-related parameters while minimizing false alarms.
2. **Data Collection and Aggregation:**
   - The sensors continuously collect data such as temperature fluctuations, humidity levels, and smoke presence. This data is sent to a central hub for real-time processing.
   - Additional data sources like satellite imagery and drone feeds are integrated for broader coverage and increased reliability.
3. **Data Processing and Fire Detection Algorithm:**
   - The data collected is processed using advanced machine learning algorithms, which are trained to detect early signs of wildfire. The algorithm discerns between actual fire events and non-fire phenomena like fog, dust, or sunlight reflections.
   - This step ensures that the system responds only to actual fire risks, improving efficiency and reducing false positives.
4. **Alert and Notification System:**
   - Once a potential fire is detected, the system triggers an automatic alert to local authorities, firefighters, and forest management teams.
   - The alerts are sent through multiple channels, including SMS, emails, and app notifications, ensuring that the response team is quickly informed.
5. **Visual Monitoring and Decision-Making:**
   - A real-time dashboard is used by decision-makers to monitor the current status of the forest, providing a live map of the areas being monitored and any detected fire outbreaks.
   - Visualizations include sensor data and predictive fire-risk zones based on weather conditions and historical fire data.
6. **Response and Mitigation:**
   - Based on the detected fire and the alert notifications, response teams are dispatched to the site.
   - The system continuously monitors the situation and provides updates on the fire’s spread, wind conditions, and risk factors, helping responders adjust their strategy in real time.



3: DOMAIN MODEL SPECIFICATION (Anushka Karhadkar)

1. **Sensors:**
    - Attributes - ID, coordinates, operational status (on/off), type (thermostat, smoke, dampness).
    - Description - Refers to the various sensors that have been set up within the forest for the purpose of collecting data regarding the various conditions of the environment.
    - Relationship - Belongs to fire detection and sensor network elements.

2. **Sensor Network:**
    - Attributes - status of the network, number of sensors, operational area and ID
    - Description - This is the set of sensors that work on different forest environmental conditions.
    - Relationship - Provides data for the Central Monitoring System

3. **Central Monitoring System:**
    - Attributes - System ID, Live Status, Data storage
    - Description - The system of a cost–efficient sensors with different responsibilities to observe aspects in the forest environment.
    - Relationship - Feeding data to the Central Monitoring System

4. **Fire Detection Algorithm:**
    - Attributes - Algorithm version, accuracy rate and false alarm rate.
    - Description- This one processes the sensor information and pictures by AI/ML strategies that may assist in identifying the presence of wildfire.
    - Relationship - Uses a Central Monitoring System for multi-source enablement

5. **Weather data:**
    - Attributes - Weather ID,temperature,wind speed, humidity
    - Description - This is the actual data of weather conditions to estimate and evaluate the forest fires , its presence and also extent.
    - Relationship - Used in data analysis to predict certain data based on past weather conditions.

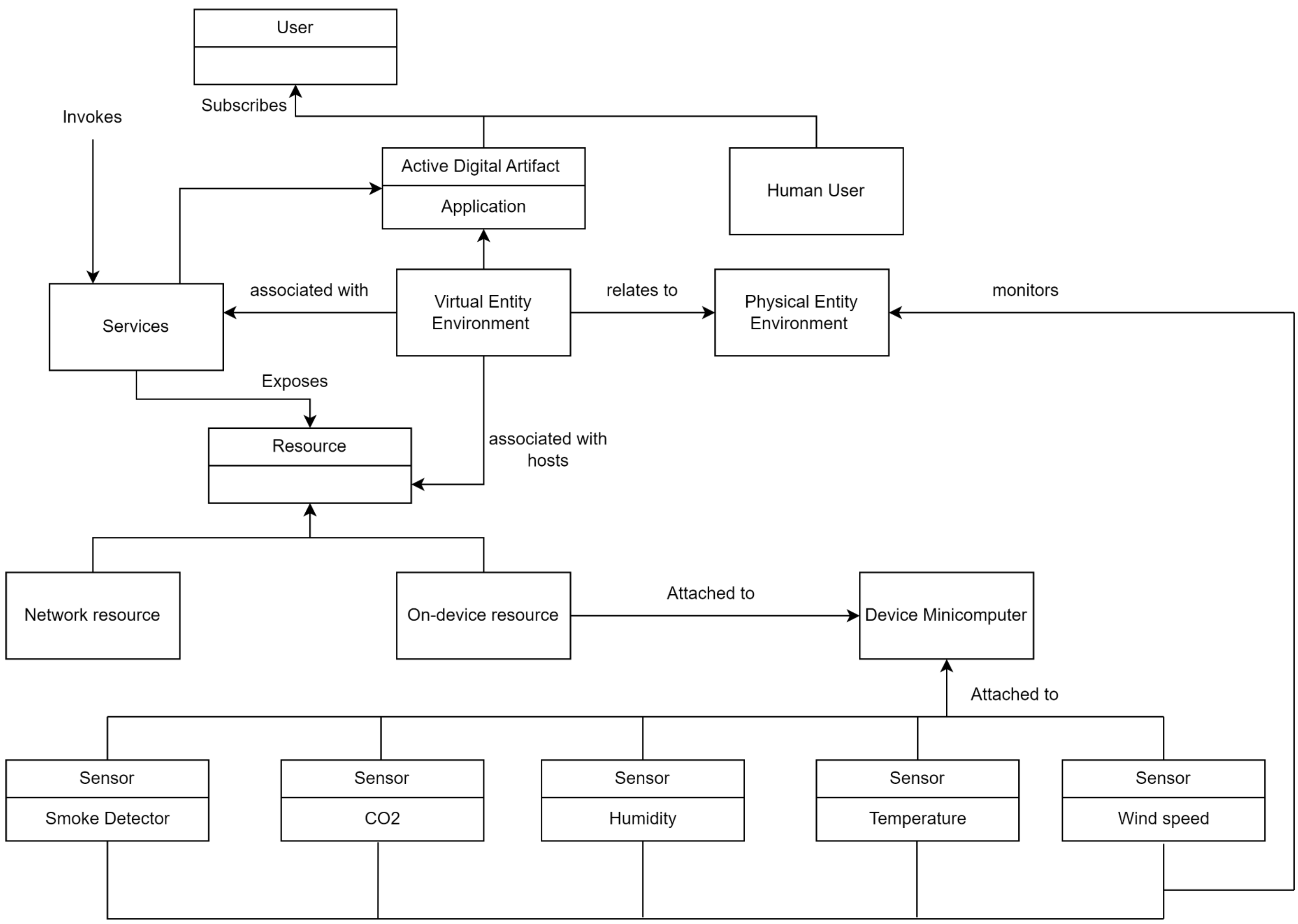
6. **Drones:**

Attributes: ID, Battery consumption status, area under observation

Description: Drones to collect visual data like images, video footage to further analyze the area and help in detecting fires.

Relationship: They provide additional visual data to the Central Monitoring System.

Diagram :



4: INFORMATION MODEL SPECIFICATION (ANIKET MAHAJAN)

The information model of the IoT-based forest fire detection system is the structural outline forming the basis of different forms of data exchanged amongst various components of the system. It is mostly concerned with the attributes and relationships of virtual entities, thus ensuring effective data handling without going into specifics of how that data is expressed or stored.

The basic concepts of the information model include virtual entities like environmental sensors and drones. The main sensors are the DHT22, to measure humidity, the MQ7 sensor for smoke levels, and LM35 temperature sensor, represented by attributes like type of sensor, measurement units, and operational status. For example, DHT22 records humidity percentage, while MQ7 collects info on smoke concentration, and these are crucial to fire detection. Such attributes assist in proper assessments of environmental conditions that may lead to forest fires.

The model includes relationships between entities surrounding the interconnectivity of the drones and the ground sensors. Cameras and thermal sensors aboard the drones uniquely supplement data presented by stationary sensors. By having aerial surveillance, drones duly supplement the entire monitoring capability of the system thereby enhancing its likelihood to provide proper assessment of potential fire threats.

Generally speaking, concerning data flow, the information model describes how sensor readouts are likely sent to a Raspberry Pi, which in turn works as an edge device with respect to the further segue of this data into AWS IoT Core for basically other processing and analysis, not forgetting centralized provision and management of information. The model also specified those events which are attributes related to fire detection, which trigger set thresholds for alarms. For example, this may be regarded as an alert when the readings of temperature from the LM35 exceeds the predetermined limits, indicating a risk of fire.

That said, this information model really constitutes an underlining blueprint, providing guidance for interoperability and the integration of data in order to enable effective monitoring and warn-off.

5: SERVICE SPECIFICATIONS. . (Kaushik)

A forest fire detection system involves multiple services in harmony to detect fires early and trigger necessary responses. These services operate across multiple layers, including data collection, analysis, decision-making, and alert systems. Below is a breakdown of the key services involved:

i. Sensor Network Management Service

This service handles the deployment, calibration, and real-time monitoring of various sensors (e.g., temperature, humidity, smoke). It ensures sensors are functioning correctly and communicates sensor data to the central system.

ii. Data Aggregation and Integration Service

This service collects data from various sources such as satellite imagery, drones, and ground sensors. It integrates this data into a unified format for further analysis. It is responsible for syncing real-time weather data, wind patterns, and environmental parameters to ensure comprehensive monitoring.

iii. Fire Detection Algorithm Service

This service utilizes AI/ML algorithms and image processing techniques to detect early signs of wildfires. It processes the aggregated data to identify fire patterns, eliminating noise from fog, dust, and other false alarm factors.

iv. Alert and Notification Service

When a fire is detected, this service sends alerts to the relevant authorities (forest officials, firefighters, local communities) via SMS, email, or mobile apps. It also integrates with emergency response systems to facilitate rapid deployment of resources.

v. Visualization and Monitoring Dashboard

This service provides a graphical interface for monitoring forest areas in real time. It visualizes sensor data, fire locations, and risk areas on a map, allowing stakeholders to make informed decisions quickly.

vi. Predictive Analytics Service

This service forecasts potential fire risks based on historical data, current weather conditions, and fire patterns. It helps predict the likelihood of fires in certain areas, allowing preventive actions to be taken.

6: IoT LEVEL SPECIFICATION (Priyank Kulkarni)

It comes under IoT Level 3 deployment, as it gathers real-time data in a location and processes them at the edge.
Sensor Deployment (Perception Layer):
Humidity, smoke, and temperature sensors shall be installed to gather the critical environmental data required to detect fire
Localized Processing using Edge Computing:

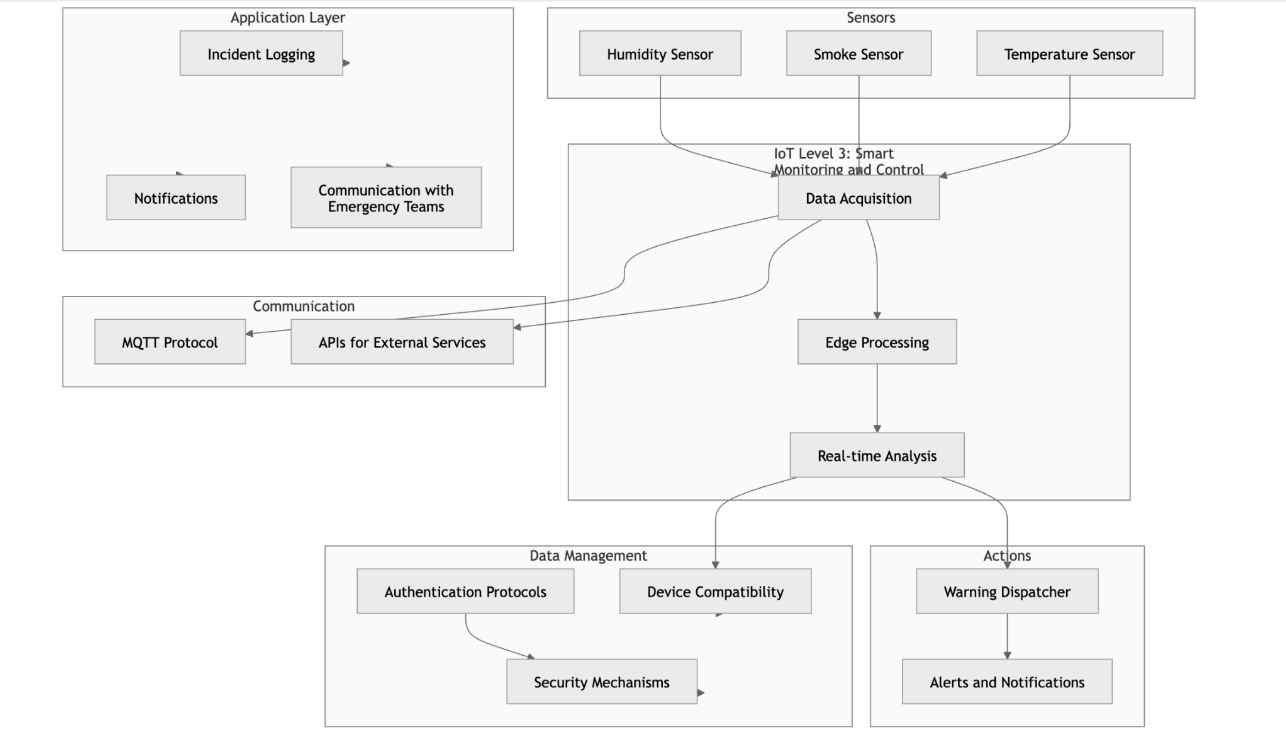
Edge computing-enabled end devices allow processing at local levels using sensor data. Decision-making can be performed in real time as local analysis is instantaneous, and it's not dependent on cloud-based systems.
Data Relay to Control Centers using MQTT Protocol
The system makes use of the MQTT protocol that relays data from sensors to control centers in minimum latency. API for out-scaling external services:

APIs offer an explicit interface towards the outside services from which the control center is out of control, similar to the fire department. In this way, it simplifies swift communication and coordination in case of an emergency.
Software Actuators for Alarms
Software actuators are utilized for the purpose of sending a warning and notification whenever sensor data exceeds pre-set threshold values (for example, high temperature or smoke detection). Data Management and Device Compatibility:

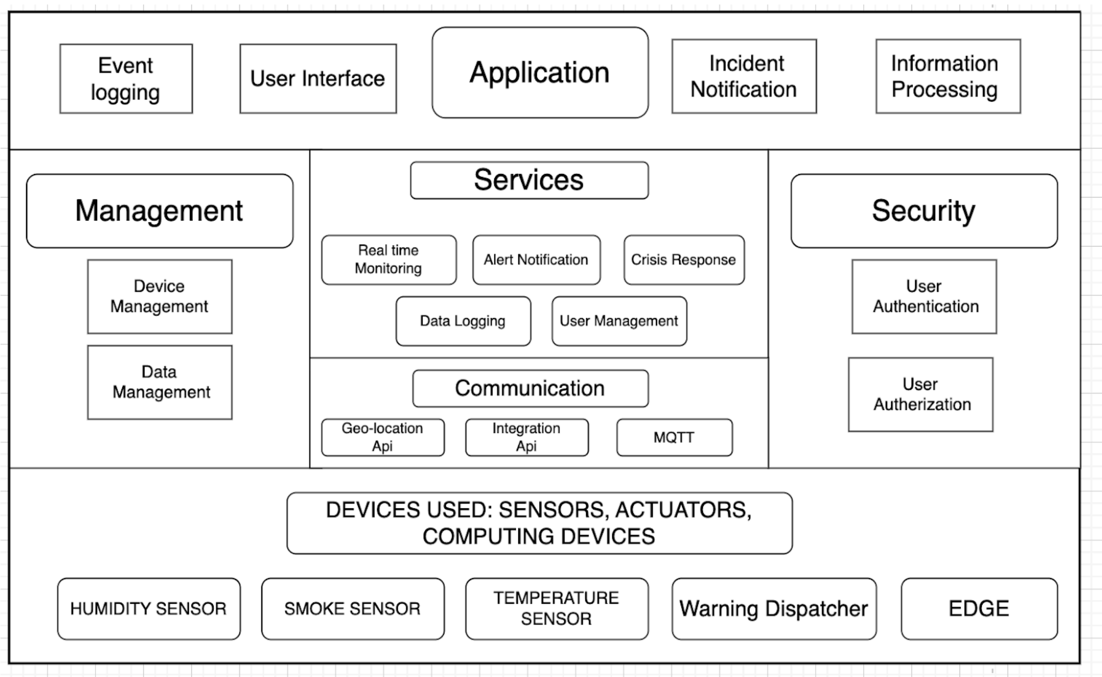
The system will have advanced data management capabilities and compatibility for devices, allowing for fluent operation with easy third-party services for fantastic functionality.
Security Mechanisms:

The key core security mechanisms inside the system are supported with authentication protocols, which ensure that no one penetrates into the system without authority and ensures proper secure operation.
Application Layer Functionalities:

The application layer offers some critical functionalities such as
Logging of all incidents: Automatic logging of all events for analysis and reporting.
Notifications: instant alerts on the occurrence of a fire incident or hazardous conditions.
Communication to Emergency Teams: It will facilitate direct communication with emergency teams to act in concert.
Proactive Fire Management:
In the third level, the IoT system will guarantee fast on-site responses in fire-prone areas though one is maintaining real-time monitoring.
Cloud Integration: This allows for flexibility, thus, integration of cloud services for further data processing and storage where it's needed.

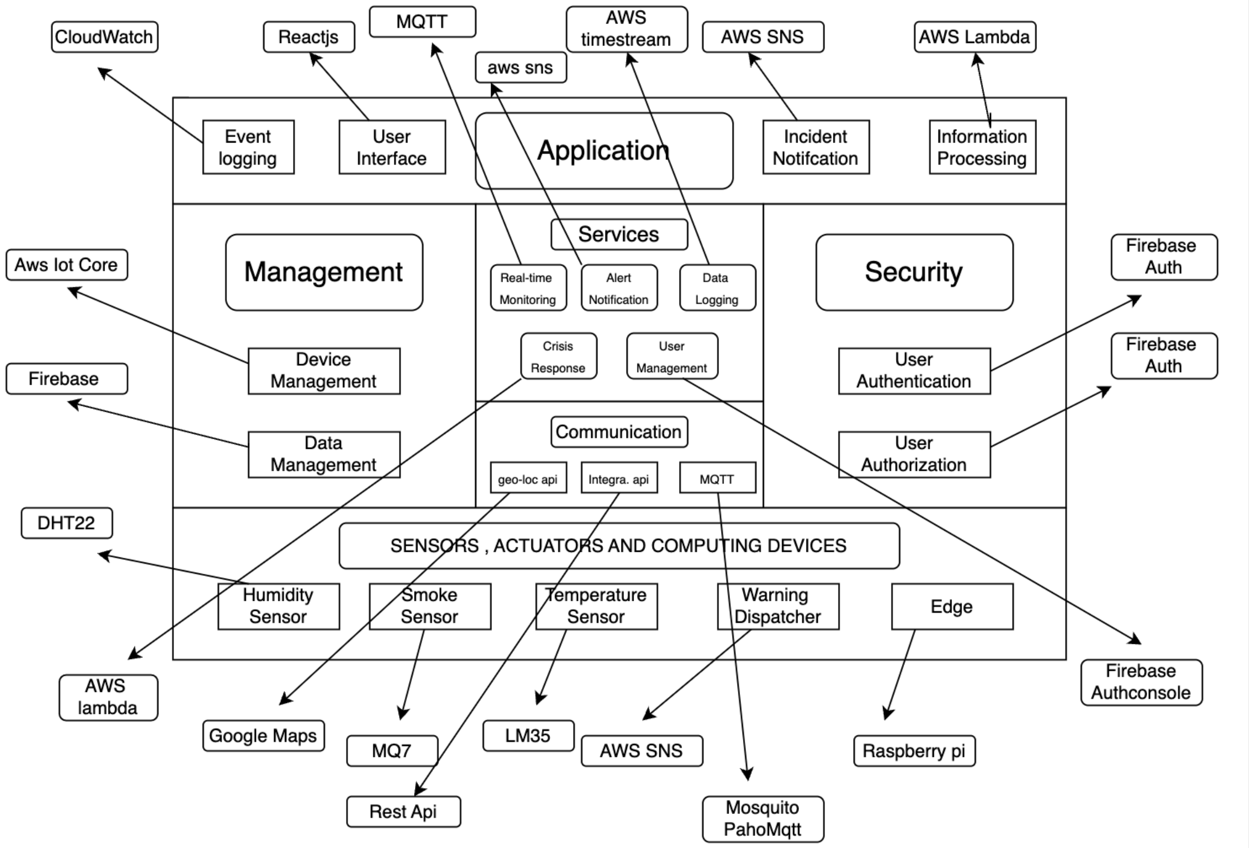

Application Layer
Incident Logging
Notifications
Communication with Emergency Teams
Sensors
Humidity Sensor
Smoke Sensor
Temperature Sensor
IoT Level 3: Smart Monitoring and Control
Data Acquisition
Edge Processing
Real-time Analysis
Communication
MQTT Protocol
APIs for External Services
Data Management
Authentication Protocols
Device Compatibility
Security Mechanisms
Actions
Warning Dispatcher
Alerts and Notifications

7: FUNCTIONAL VIEW SPECIFICATION (Tejaswi Koul)



This diagram presents the functional view specification for our forest fire detection system , which detects and reacts to the possibility of forest fires . Firstly, there are devices placed at the bottom of the hierarchy which include humidity sensors, smoke sensors, temperature sensors and edge computing devices. This class of sensors are critical in the acquisition of near real time data for environmental parameters such as moisture in the air, presence of smoke and temperature fluctuations which indicate potential threats of forest fires in the forests. Another category is the actuators ,where we consider a warning dispatcher , which is a software actuator, as it generates alerts and notifications when fire detection thresholds are met. Edge computing devices carry out the processing of the sensor data at the local site such that there is quick analysis and responses without reliance on cloud systems which enables intervention operations to be timely.The Use of communication layer incorporates MQTT and geo-location based application programming interfaces to relay possible fire outbreak coordinates to a control center. Integration APIs allow for exchange of information between system components and third party services e.g local fire departments etc.The Services layer provides real-time supervision, alerting capabilities and a way to react in case of a fire incident. It also includes data management and access control features for seamless operations and expansion. The Management section is concerned with the management of data and devices with a focus on the handling of the sensor data and managing device compatibility. The Security layer deals with protection of the system from unauthorized access by providing authentication. The topmost layer of the forest fire detection system functional view is the application layer . This layer provides an essential user functionality which includes the user interface, event logging, incident notifications among other factors. It includes sensor information and other services to prevent and detect fire and evaluate the risks. This layer enables the efficient handling of communication of the emergency team as well as the general public, thus, limiting the risks of fire causing incidents.

8: OPERATIONAL VIEW SPECIFICATION (TEJASWI KOUL)



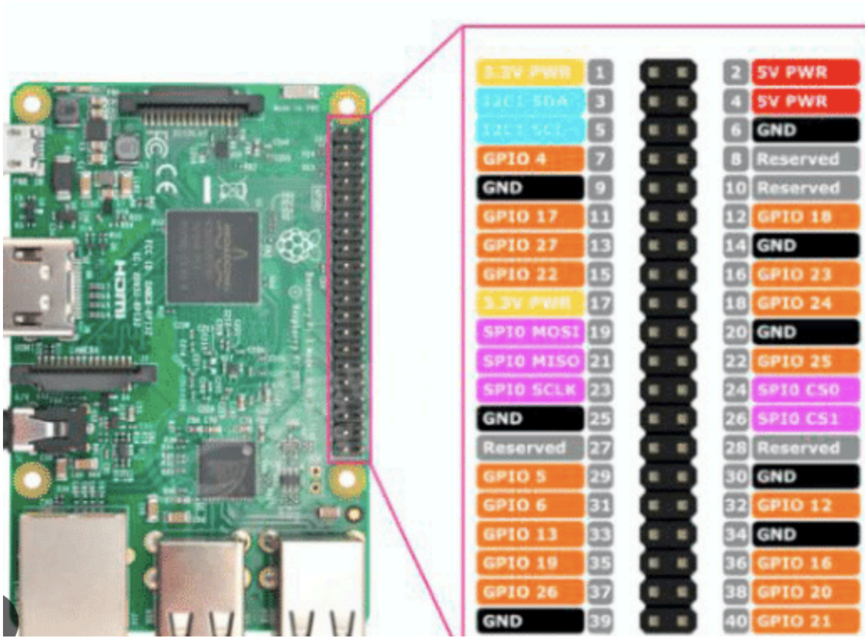
The operational view design of this IoT-based forest fire detection system has four important layers known as Sensors/Actuators, Management, Application and Security. They work together so as to facilitate real time monitoring, incident alerting and flow of information in a secure manner. In the Sensors/Actuators layer we have devices such as the DHT22 humidity sensor, MQ7 smoke sensor, the LM35 temperature sensor, which collect data about the environment. This data is processed in connection with a Raspberry Pi that acts as an edge device and transmits sensor values in real time over MQTT .There are drones with cameras and thermal sensors that also travel around for purposes of monitoring the area and feeding the system with more aerial data for an even larger scale fire detection system.

Device and data management for this layer is provided by Firebase and AWS IoT Core. AWS CloudWatch is used to log events in order to monitor the performance and usage of the system. The Application layer provides crucial support for services such as instantaneous monitoring, emergency management, and warning services. Sensor data is stored through AWS Timestream for future research purposes ,while in case of fire alerts, AWS SNS is used to send out the alerts to any mobile devices or emails found in the database. The user interface is developed in React.js, which is the front-end part of the system where end-users can retrieve and input any needed information.

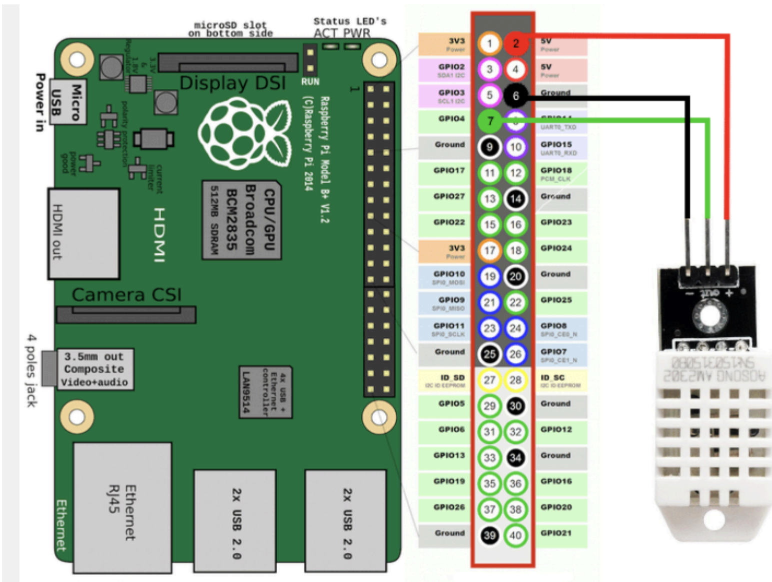
To restrict access to the system, the Security layer uses Firebase Auth for handling the user's authentication and authorization requirements. The architecture provides effective communication and data handling as well as prompt action to emergencies, thanks to the use of cloud technology, while drones aid in monitoring and tracking fire in real-time .

9: DEVICE AND COMPONENT INTEGRATION (ANKET)

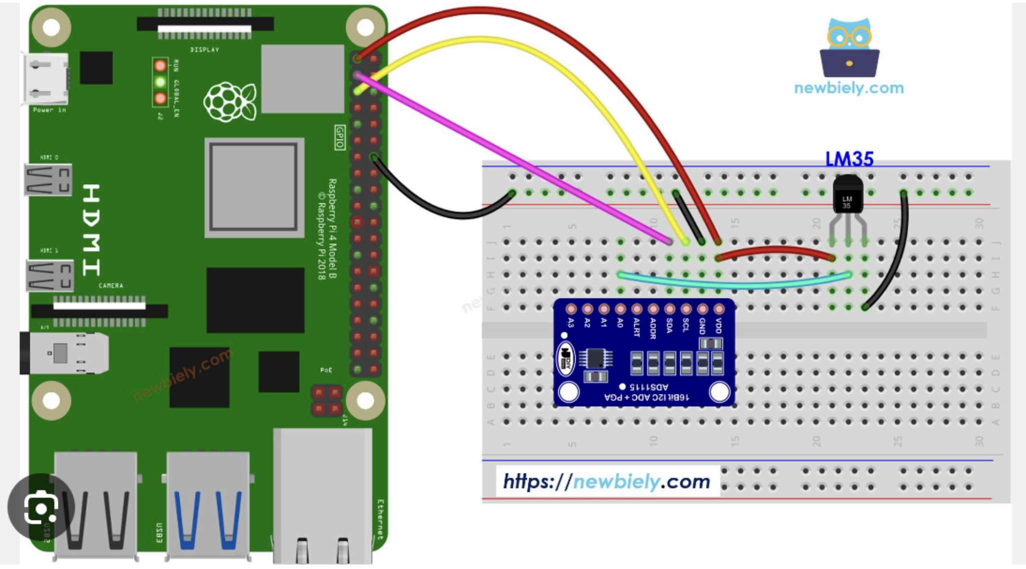
1. RASPBERRY PI



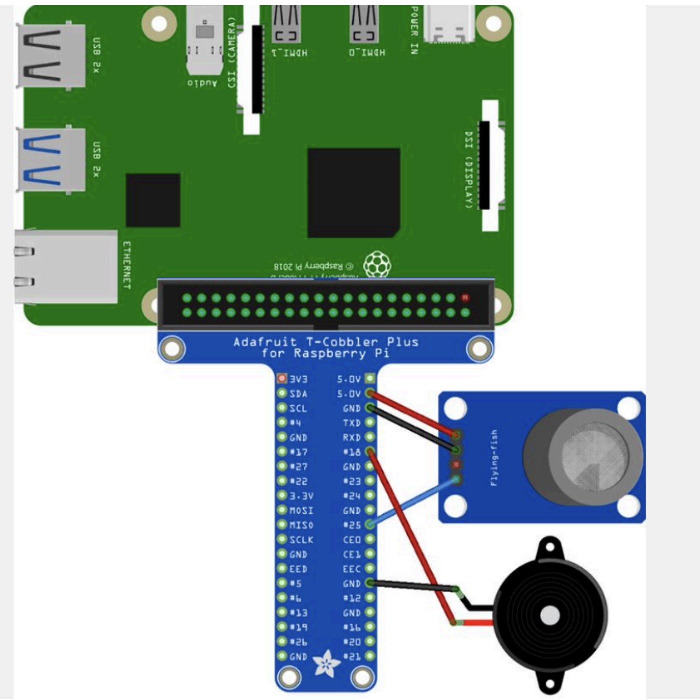
2.DHT22



3. lM35


newbiely.com
LM35
https://newbiely.com

4.MQ7


Adafruit T-Cobbler Plus
for Raspberry Pi

10: APPLICATION DEVELOPMENT (Payoshni Khekale)

The application development phase is where all the technology comes together to create a system that's not only powerful but also easy to use for those who need it most—like forest rangers and firefighters. Here's how the application was built:

1. **Choosing the Right Tools:**
   - We used a variety of tools to make sure the system works seamlessly. For instance, Raspberry Pi devices were placed in the field to collect data, and we used AWS IoT Core in the cloud to process it all. The real-time communication happens through the MQTT protocol, which ensures data flows smoothly between the sensors and the cloud.
   - The frontend—the part of the system people interact with—was built using React.js to create an intuitive dashboard. For the backend, we used Node.js and Express.js to manage all the communications behind the scenes, while MongoDB stores the data for future reference.
2. **Creating a User-Friendly Dashboard:**
   - We knew the system had to be simple and effective, so we designed a real-time dashboard that lets users see data from the forest sensors—like temperature, humidity, and smoke levels—on an interactive map. If there's any indication of a fire, the dashboard immediately shows an alert, pinpointing the location and severity of the risk.
   - Users can also review historical data, track patterns, and adjust the system settings when necessary, making it a powerful tool for forest management.
3. **Building the Brains Behind the System:**
   - On the backend, we built the logic that brings everything together. The system gathers data from various sources, including ground sensors and even drones. We created a RESTful API to handle the communication between the sensors, the cloud, and the user interface.
   - Our fire detection relies on machine learning models that analyze the incoming data and assess the risk of a fire. This predictive approach helps us act before things get out of hand.
4. **Going Mobile for On-the-Go Updates:**
   - We also developed a mobile app for users like firefighters, who need real-time updates in the field. This app provides quick access to essential data like the fire's location, wind conditions, and weather updates, all from their phones. Plus, when a fire risk is detected, the app sends push notifications, ensuring that everyone is informed immediately.

5. **<u>Working with Other Systems:</u>**

    - The application isn't just a standalone system. We integrated it with local fire departments' systems, allowing them to receive live updates and even dispatch firefighting teams automatically. Additionally, we linked the system with weather services, so it can predict areas at risk based on factors like wind speed and humidity.

6. **<u>Testing and Rolling It Out:</u>**

    - Before launching, we ran extensive tests to ensure everything worked correctly, especially the accuracy of the fire detection models and the reliability of real-time updates. Once the system passed all our tests, we deployed it on AWS Cloud, making it scalable and easily accessible for everyone involved.

7. **<u>Listening to Users and Improving Over Time:</u>**

    - Even after deployment, the system is designed to evolve. We regularly collect feedback from users to improve the interface and add new features as needed. Over time, we plan to integrate even more advanced technology, like drone surveillance, to make the system even more effective at protecting forests.