IoT APPLICATIONS FOR SMART ENVIRONMENT AND AGRICULTURE

Smart Environment-Monitoring

Environment monitoring refers to actions that are required for characterising and monitoring the quality of the environment. A smart environment monitoring system should enable the following:

1. Preparations for assessment of environment impact
2. Establish the trends in environmental parameters and current status of the environment
3. Interpretation of data and evaluate environmental quality indices
4. Monitor the air, soil and water quality parameters

Monitor harmful chemicals, biological, microbiological, radiological and other parameters ***Weather Monitoring System***

A smart weather monitoring system should enable the following:

1. Each measuring node for weather parameters is assigned an ID. Each lamppost deploys a wireless sensor node. Each node measure the T, RH and other weather parameters at assigned locations. A group of WSNs communicates using ZigBee and forms a network. Each network has an access point, which receives the messages from each node.
   They depicted interconnections between nodes, coordinators, routers and access points. Each access point associates a gateway.
2. The nodes communicate the parameters up to the access point using WSNs at multiple locations.
3. Forward and store the parameters on an Internet cloud platform
4. Publishes weather messages for the display boards at specific locations in the city and communicates to weather API at mobile and web users
5. Publishes the messages in real time and send alerts using a weather reporting application
6. Analyse and assess the environment impact
7. Enables intelligent decisions using data and historical analytics reports at city cloud weather data store

Two domains and their high-level service capabilities in the weather monitoring services in IoT architecture reference model are:

1. **Device and Gateway Domain:** Assume that the system deploys m weather-sensor embedded devices, each with a location-data sensor and n access-points for the WSNs. A sensor node does minimum required computations, gathers sensed information and communicates

with other connected nodes in the network.main computations and puts the result in real time updated database. The items identified for communication from gateway are queried from the database. The items communicate from gateway using network protocols and HTTP/HTTPS services.

2. **Device subdomain**: Hardware WSN board consists of sensors for weather parameters. A board example is Waspmote. Following are the sensor circuit features: ultralow power dissipation; multiple transceiver interfaces, such as ZigBee and Wi-Fi (for medium range), RFID, NFC, Bluetooth 2.1 or BL LE (for short range) and LPWAN, 4G, 3G (for long range); OTA programmability, AES, RSA, MD5, SHA, Hash (as encryption libraries) and bus protocols, such as CAN and RS232C.

3. **Gateway Subdomain:** The parameters and alerts communicate to a local or remote web service, time-and location-stamping service, item provider, protocol bindings and 6LowPAN/IPv6 modules as per configuration setting at the configuration administration service of OSGi framework. The bindings between ZigBee LANs, 6LowPAN and LPWAN and IPv6 protocols are used for networking of the devices, WSNs, OSGi with the HTTP/HTTPS services.

4. ***Application and Network Domain***: Applications and network domain deploys the applications and services and has high-level capabilities, such analytics, data visualisation, display-board feeds, weather reporting application, and IFTTT triggers and actions. The cloud platform can be IBM Bluemix, AWS IoT or TCUP.

Domain Architectural Reference Model

It layered architecture for network of nodes with fixed connecting infrastructure and the mobile WSNs using coordinators, relays, gateways and routers. Figure depicted a layered architecture for network of nodes with multilevel hops and router or gateway and access point. Figure shows the data-flow diagram and domain architecture reference model for the WSN based monitoring services.

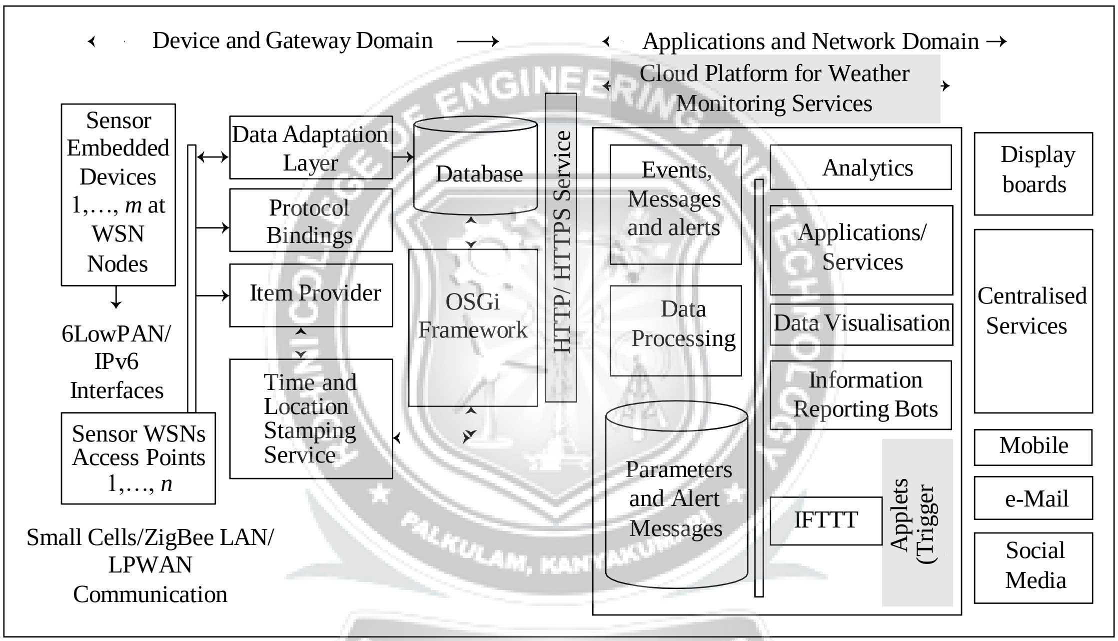

Figure Data-flow diagram and domain architecture reference model for the WSNs based monitoring services

Devices Hardware Design and Code Development Environment, Development, Debugging and Deployment

A microcontroller circuit consists of memory, over the air programmability (OTP) and transceiver associated with each sensor or node. The weather monitoring circuit deploys sensors for T, RH and atmospheric pressure (P_{atm}) and may include solar visible radiation, wind speed and direction, and rainfall.

Hardware design of the sensor and WSN node can use Arduino board with ZigBee shield. Alternatively Waspmote, an open source wireless sensor platform for autonomous wireless low power including WSN nodes, called MOTEs (mobile terminals). Waspmote node battery power can be between 1 year and 5 years depending on the transceiver, frequency and the radio used. The platform is compatible with the Arduino IDE and has a community support.

The edge devices and WSNs codes development uses IDE such as the Arduino or Eclipse IDE for Java Developers.

Weather Reporting Bot

A bot is an application that runs automated or semi-automated scripts a for specific set of tasks and communicates the results over the Internet.[25] A bot generally performs the task which are simple and structurally repetitive, such as a, weather reporting bot. The word 'bot' is derided from the word *robot*.

A bot can communicate with an API using Instant Messaging (IM) or Internet Relay Chat (IRC) or to Twitter or Facebook. A bot can also chat and give responses to the questions from user API.

A widely used bot is an application in which the script fetches, analyses and files information from a web server. A server may specify in a text file, called robots.txt for rules on bot behaviour on that server. A server can use software to deny access in case the script does not observe the rules specified at the file.

A weather bot is multitasking. That can be used to communicate a report on a mobile. The bot fetches, analyses and communicates information to a report seeking API. The bot uses the weather parameters and generates the alert messages from the database and messages for forecast by a cloud analytics service.

A mobile app can display the report in two succeeding frames repeatedly. The first frame shows the weather condition of the current day, as:

1. First line: condition such as clear, rain, partly rain, cloudy or partly cloudy
2. Second line, first part text gives the day current *T* and four or five spaces
3. Second line, second part text gives superscripted text for the maximum *T* expected and subscripted text for minimum *T* expected, followed by four or five spaces
4. Second line, third part text gives superscripted text for current RH% value and subscripted text for wind-speed in kmph (kilometer per hour).

Thus, first reporting frame displays the current condition and day's forecast for T_{max} and T_{min}. Second frame shows the forecast for today, tomorrow and day after for weather as:

1. First line: "Sat Sun Mon"
2. Second line shows a symbol which is completely unfilled circle for sunny, or cloud image with sun for partly cloudy, or cloud sign for fully cloudy below each day

Sat, Sun and Mon. After the sign, a superscripted word gives maximum *T*, and a subscripted word, the minimum expected on that day.

Thus, forecast for three days is reported, viz. today, tomorrow and day after.

Example of creating a weather bot is Slackweatherbot API.[26] The bot uses the codes given a Farnciskim site.[27] The API is a node.js module for the bot. It displays Second frame shows the forecast for today, tomorrow and day after days for weather, for example as follows:

1. First line: Bot name Current Runtime (such as 09:15 A.M.)
2. Second line shows “Condition for *City Name*, *Place Name*, *Current Time*, *Standard* (suchas IST, GMT)
3. Third line shows “Today (such as SAT): *T*, *condition* (such as sunny, partly cloudy,cloudy or rain)”
4. Fourth line shows “Tomorrow (such as SUN) Current: *T*, *condition*”
5. Fifth line shows “Day after (such as MON) Current: *T*, *condition*”

Air Pollution Monitoring

A growing problem for all residents is air pollution from cars, toxic gases generated in factories and farms, such as carbon monoxide (CO). Pollution needs monitoring and to ensure the safety of workers and goods inside chemical plants. The monitoring does the following tasks:

1. Monitoring and measuring levels of CO, a gas dangerous above 50–100 ppm level; carbon dioxide (CO_2), a gas causeswhich greenhouse effect; and ozone (O_3), a gas dangerous above 0.1 mg/per kg air level, for controlling air pollution
2. Monitoring and measuring levels of hydrogen sulfide (H_2S), a highly toxic gas. It is a greenhouse gas so its increase may contribute to global warming as well.
3. Monitoring and measuring levels of hydrocarbons, such as ethanol, propane.
4. Measure T, RH and P_{atm} parameters for calibrations of sensed gaseous parameters of each node
5. Investigate air quality and the effects of air pollution.
6. Compute Air Quality Index (AQI) from the parameters, such as hourly or daily averages of air pollutant concentration, particulate matter (such as dust or carbon particle)
7. Compute source and spatial dispersion of pollutants as a function of day conditions, wind-speed and direction, air temperature and air temperature gradient with altitudeand topography using analytics.
8. Data visualisation
9. Report the pollution status to monitoring authorities

Sensors play a vital role in air-quality monitoring. The application has eleventh ranking among 50 sensor-applications for a smarter world.

A data-flow diagram and domain architecture reference model for air pollution monitoring services are similar to Figure. Two domains and their high-level service capabilities in the air quality and pollution monitoring services in IoT architecture reference model are:

1. *Device and Gateway Domain*: Assume that the system deploys *m* gas sensor embedded devices at each WSN with a location-data sensor and *n* access-points for the WSNs. The data-adaptation layer at gateway does the aggregation, compaction and fusion computations for each sensor node data. The queries gather sensed information from the database and the items selected communicate using HTTP/HTTPS/MPLS services.

   WSN board IO ports connect the sensors for gaseous, particulate matter and weather parameters. Each sensor node is configured by assigning a node ID. Anode ID maps with the GPS location found earlier from GPS modules at the data- adaptation layer at the gateway.

   A sensor ID is configured for each sensor at the node. Each sensor associated circuit is also configured for frequency of measurements every day and interval between two successive measurements. The sensor circuit is configured to activate only for measurement duration at a measuring instance followed by long inactive intervals. coefficients of the temperature sensor. Refer Example which gives the codes for uses of the calibration coefficients of sensed data communicated by I2C serial bus. The coefficients enable to sense parameters with enhanced accuracy.

   The coefficients for a gas-sensor output depend on the RH and P_{atm} Therefore, these parameters are also measured along with gas sensor outputs. Mapping of a sensor output with the THP coefficients at the adaptation layer will result into accuracy in the measurements of values of sensed parameters.

   An example is Waspmote board[24] which can be used with sensors such as city pollution CO, NO, NO_2, O_3, SO_2 and dust particles sensors and air-quality finding sensors for SO_2, NO_2, dust particles, CO, CO_2, O_3 and NH_3. The Arduino or Eclipse IDE can be used to develop codes for the Waspmote.

2. *The Applications and Network Domain:* The applications and network domain deploys the applications and services and have high-level capabilities, such as events, messages,alerts and data processing, databases, applications and services, analytics, data visualisation, display-board feeds, pollution reporting applications and services, and IFTTT triggers and actions. The cloud platform can be TCUP, AWSIoT, IBM Bluemix or Nimbits.

Forest Fire Detection

A big problem for countries with large forest areas is forest fires.
A fire monitoring service does the following tasks:

1. Uses OTP features for programmable WSNs and gateways
2. Measures and monitors the T, RH, CO, CO_2 and infrared light (fire generated) intensity in real time at preset intervals
3. Each WSN uploads the program and preset measured intervals of t1 (say, 300 s) eachand the preset measured intervals of t2 (say, on 1 or 5 s) on sensed parameters valuesexceeding thresholds can instantaneously trigger the fire-alarm algorithm
4. Configures the data-adaptation layers with calibration parameters
5. Communicates the WSN messages at the preset intervals to the access point associatedfor specific network area
6. Communicates alerts, triggers, messages and data at data-adaptation layer using anuploaded program at associated gateway
7. Uploads connectivity programs for gateways
8. Runs at the data-adaptation layer the faulty or inaccessible sensors at periodic intervals
9. Integrates data with the node locations found from mapping with node IDs, compute,and activate the alarms using an algorithm, input-sensed and calibrated coefficients
10. Processes the layer data and database information, and communicates instantaneouslyto nearest mobiles and fire-fighting services near the access point gateway
11. Updates the database and communicates to a cloud platform, such as Nimbits, my.openHAB, TCUP, AWS or Bluemix platform
12. Modifies the preset measured intervals to t2 on activation of the fire alarm after valuechanges above the configured threshold values
13. Uses analytics to evaluate reliability index of the preset, threshold and configurationvalues and need to update alarm-algorithm and if needs improvement then upload new algorithms
14. Uses analytics to generate and communicate topological maps for the currently fire-infected forest area and reachability maps for fire-fighting service equipments

Sensors play a vital role in forest-fire monitoring. The application has tenth ranking among 50 sensor-applications for a smarter world.

Figure shows a data-flow diagram and domain architecture reference model for the monitoring service.

The figure shows that the service deploys *m* embedded-sensor devices at each of *n* WSN associated with *x* access points. Device and gateway domain functions in the fire monitoring service for forests in IoT architecture reference model is as follows:

A lookup table enables mapping of two entities. Location-data stamping uses sensorIDs at a lookup table. Data adaptation of each sensor is at the layer. Data aggregates,compacts and fuses, computes, gathers sensed information and the algorithms use that for alarm and faulty sensor identification and configuration management. Datastore at the database, updates in real time. The alerts and messages communicate toIoT cloud platform.

Hardware WSN board and sensors can use Waspmote board. Each WSN communicates to access points using a multiprotocol wireless router.

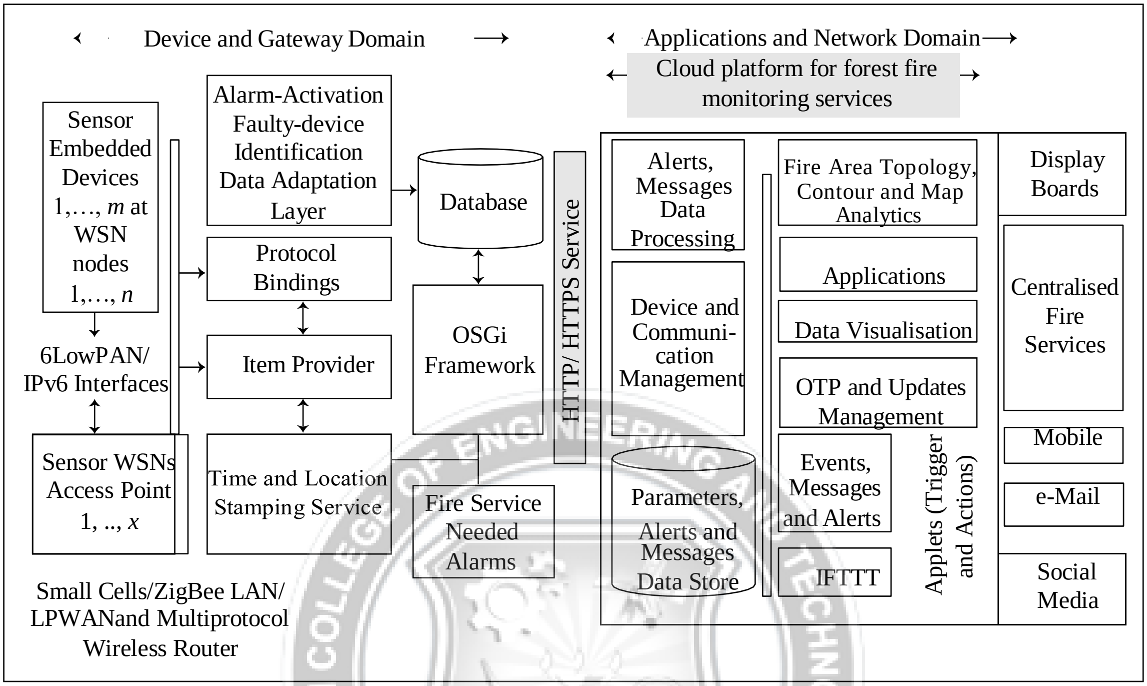

Figure Data flow diagram and domain architecture reference model for the WSNs based ForestFire monitoring service

Smart Agriculture

Following section describes two applications viz., smart irrigation in crop fields and smartwine quality enhancing.

Smart Irrigation

Smart irrigation deploys sensors for moisture. A smart irrigation monitoring service does the following tasks:

- Sensors for moisture and actuators for watering channelsare used in smart irrigation.
- Uses soil moisture sensors with a sensor circuitry boardwith each one installed at certain depth in the fields.
- Uses an array of actuators (solenoid valves) which are placed along the water channels and that control deficiencies in moisture levelsabove thresholds during a given crop period.
- Uses sensors placed at three depths for monitoring of moisture in fruit plants such asgrapes or mango, and monitors evapotranspiration (evaporation and transpiration)
- Measures and monitors actual absorption and irrigation water needs
- Each sensor board is in a waterproof cover and communicates to an access pointusing ZigBee protocol. An array of sensor circuits forms a

- Access point receives the data and transfers it to an associated gateway. Data adaptsat the gateway and then communicates to a cloud platform using LPWAN.
- The cloud platform may be deployed such as Nimbits, my.openHAB, AWS orBluemix.
- Analytics at the platform analyses the moisture data and communicates to theactuators of water irrigation channels as per the water needs and past historical data
- Measurements at the sensors are at preset intervals and actuators activate at analysedrequired values of the intervals.
- The platform uploads the programs to sensors and actuators circuitry and sets preset measurement intervals of T1 (say, 24 hour) each and the preset actuation interval of t2 (say, on 120 hour)
- Sensed moisture values when exceed preset thresholds then trigger the alarm
- An algorithm uploads and updates the programs for the gateways and nodes.
- Runs at the data-adaptation layer and finds the faulty or inaccessible moisturesensors at periodic intervals
- Open source SDK and IDE are used for prototyping the monitoring system.

Smart Wine Quality Enhancing

The sensors monitor the soil moisture and trunk diameter in vineyards. The monitoring controls the sugar content in grapes and health of grapevines.

Data-flow diagram and domain architecture reference-model for the monitoring serviceare similar to ones shown in Figure.

Device and Gateway Domain

A WSN measures moisture and other parameters and has an ID. Each node is a WSN. EachWSN measures at assigned places in a crop or vineyard at certain depth(s) inside the soil.Sensors at three equally spaced depths are used for the vineyard grapes sugar-control. A group of WSNs communicate among themselves using ZigBee and form a network. Each network has an access point, which receives the messages from each node using LPWAN. Figures show the WSNs. They show interconnections between nodes, coordinators, routers and access points. Each access point associates a gateway. Each gateway communicates to the cloud using LPWAN.