IoT APPLICATIONS FOR SMART AUTOMATION

Smart homes, smart cities, smart environment monitoring and smart agriculture are illustrative examples of IoT applicationsand services. Subsections below describe design approach fordesigning the projects in these areas.

Smart Home

Section 1.7.2 introduced the smart home concept and listed a number of smart home services, such as home lighting control, control and monitoring of appliances, security, intrusion detection, video surveillance, access control and security alerts, Wi-Fi, Internet, and remote cloud access, control and monitoring.

Smart-home Devices Development and Deployment using an Open-source Software

All smart home devices can communicate using openHAB.[11] HAB drives from Home Automation Bus. The developer deploys Java and OSGi services and uses the open source development environment and deployment platform. Its accompanying cloud platform[12] my.openHAB provides communication between that with the cloud. The my.openHAB cloud-connector also provides REST and cloud-based services, such as IFTTT . The operating system versions Android 4.1 onwardsand iOS7 onwards for IFTTT, enable services such as smart home controls and automation using mobile phones or tablets. OpenHAB computing environment is Java. GUI clients are designed and can be used as downloads from git.[13] IDE, guidelines, bindings forcode development are provided for openHAB.[14] It shows architectural layers inopenHAB development environment.

A *service* in figure refers to service capabilities, which can be called upon whenever needed. The figure shows the following:

1. Core openHAB objects—REST service and repository; base library
2. openHAB add-on objects—Item provider, protocol bindings, automation logics, userinterfaces and libraries
3. OSGi framework services—Configuration administration, event administration service, declarative services, log back, runtime and HTTP services

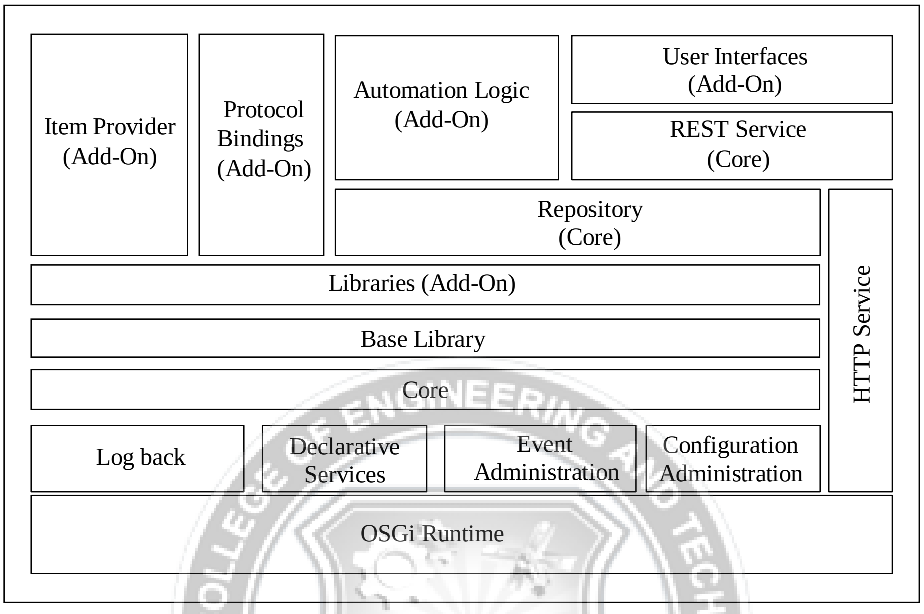

Figure Architectural layers in openHAB development environment

4. OpenHAB deploys event administration service of OSGi with pub/sub mode.
5. A stateful repository is for querying and for use by automation logics. Some functions are stateless and do not depend on previous action(s). Remaining actions are statefuland dependent on previous chain of actions. State of items in repository is as per the actions.

Two domains and their high-level service capabilities in the home automation system in IoT architecture reference model are:

1. *Device and Gateway Domain*: Assume that the system deploys *j* lighting devices, each with a proximity sensor. *Automation logic* provides that if no change is found in proximity due to presence of person(s) then the devices switch off. Assume that the system also deploys *k* intrusion sensors and *l* appliances. Automation logic provides on intrusion, communicate trigger(s) to a local or remote web-service as per configuration setting at the configuration administration service of OSGi framework.
2. *Application and Network Domain*: Applications and network domain deploys applications and services and have high-level capabilities.

Domain Architectural Reference Model

It shows the data-flow diagram and domain architecture reference model forhome automation lighting, appliances and intrusion monitoring services.

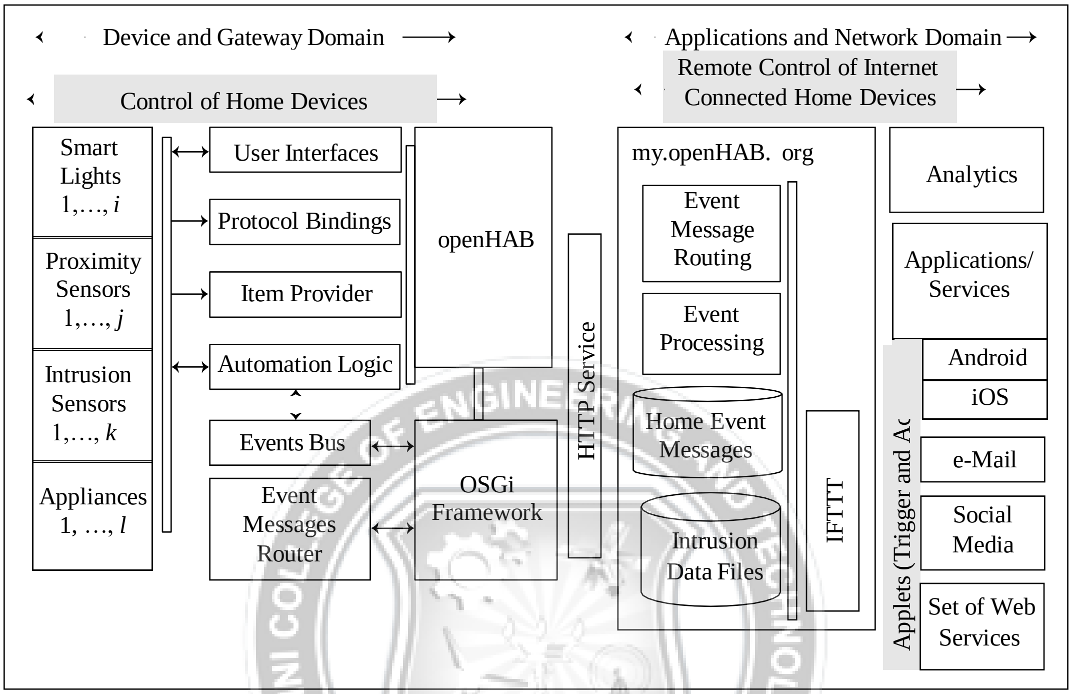

Figure Data-flow diagram and domain architecture reference model for home automation lighting, appliances and intrusion monitoring services

Figure shows that openHAB has an event bus. The bus is asynchronous. The event bus refers to a communication bus for all protocol bindings. The bindings link to the hardware. The event bus is the base service of openHAB. Example of the event is command, which triggers an action or a state change of some item or device. Another example of event is status update which informs about a status change of some item or device. For example, in response to a command.

The openHAB service is integration-hub between such devices and bindings between different protocols used for networking the home devices, OSGi and HTTP service. Usually just one instance of openHAB runs on a central coordinator (computer) at home. Event Administration Service of OSGi service is used for remote services. Several distributed openHAB instances can connect and deploy the event bus.

Code Development Environment, Development, Debugging and Deployment

The openHAB development uses the Eclipse IDE for Java Developers. Development environment can be selected as Windows (32 or 64), Linux or Mac. The Eclipse Installer adds all required plug-ins, provides encoding settings for workspace, pre-configured code formatters and automatically prepares the IDE. Workspace compilesand provides runtime by opting“ openHAB_Runtime” configuration.

The openHAB platform is neutral to hardware and interfacing protocols. For example, a security camera device may be on Raspberry platform and lighting devices on Arduino. The automation logics can connect different systems. Bus systems, devices and protocols have dedicated bindings to openHAB. Each binding User Interface (UI) design can have unique look and feel. A binding sends and receives the commands, and status updates on openHAB event bus.

The openHAB brings together different bus systems, hardware devices and interface protocols. The dedicated bindings facilitate this. The openHAB solution aims for provisioning a universal integration platform for the devices. The codes are written in Java and are fully based on OSGi. The Equinox OSGi runtime and Jetty as a web server build the core foundation of the runtime.

Devices hardware design components are 24 ¥ 7 active digital video cameras for intrusion detection, number of spatially-distributed embedded proximity sensors, the home premises; sensors data processing for detection of suspicious activities; video-processing and filtering hardware; communication network connectivity to the events bus, and on the event, communication network connectivity.

Software design modules at the device domain are software components for embedded device distributed proximity sensors data processing; filtering and extraction of events; and communication on the events, and media server gateway for communication of events.

Identifying Requirements of Network Sub-domain

Network hardware and software design components are Wi-Fi/WiMax access network, core IP network, and server. Software design components are network management functions to ensure secure communication network between device and gateway domain and applications/services.

The openHAB cloud connector connects the local openHAB runtime to a remote openHAB cloud, such as my.openHAB, instance from openHAB foundation.^{15}

Design Implementation of Device and Gateway Domain Hardware and Software

It described the sensors which can be deployed for a number of home automation applications. An implementation of home premise intrusion circuits and embedded sensor devices software needs high computing power for intrusion detection.Section 8.3.4 described Raspberry Pi 2 model B+ (RPi 2) which can be deployed due to high computing power.

An implementation of home premise lighting and appliances embedded sensor devices software needs computing power for lighting automation and Arduino or RPi boards can be deployed.

The openHAB can be used for end-to-end solutions for smart home applications and services. It described Eclipse IoT stack-based end-to-end IoT solutions with

Java and OSGi. A developer can code using Eclipse stack components for abstractions, software and gateway software in embedded sensors and devices. Sample projects using openHAB are given at Github.