Internet of Things - Unit 5

vedveethi.co.in/eNote/IoT/CS-6005 Unit 5 - Internet of Things.htm

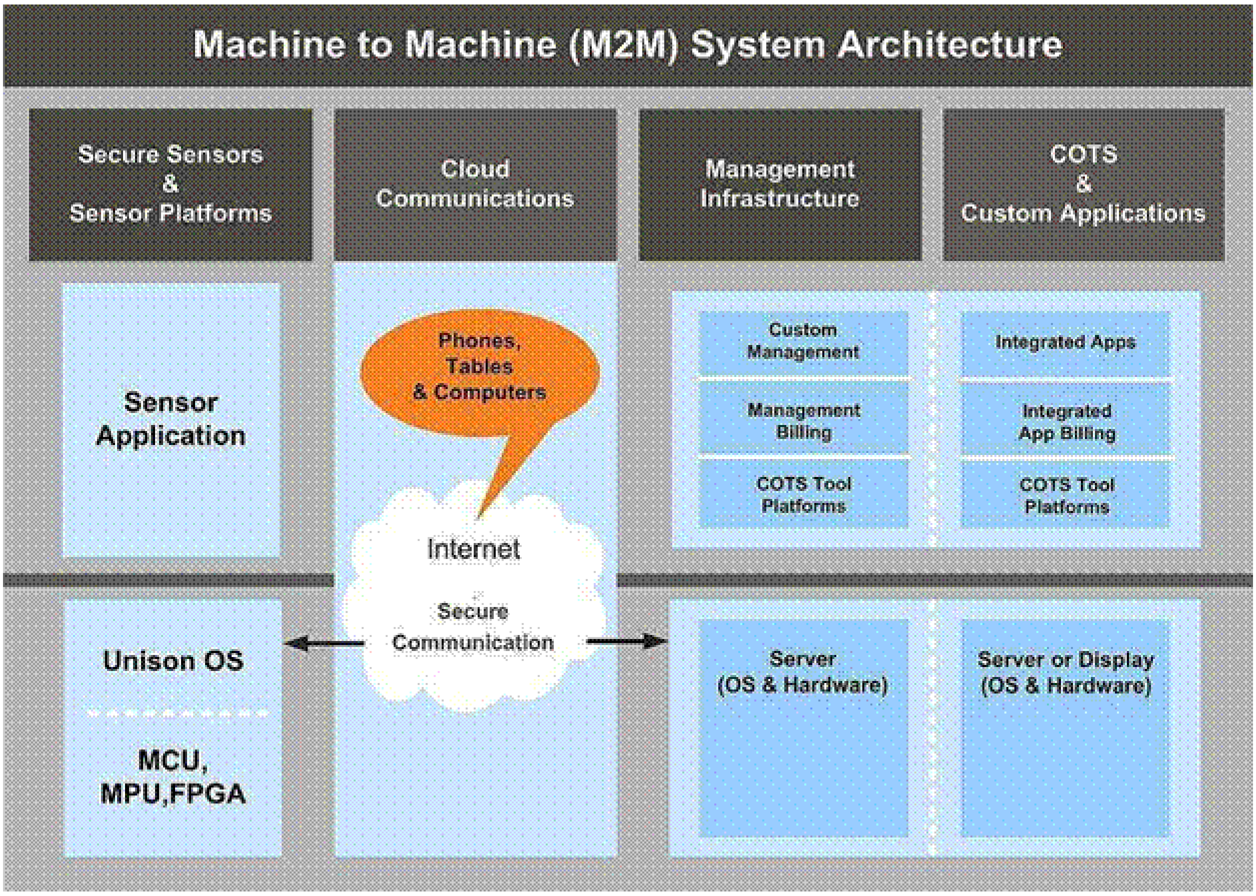

Fig .5.1 M2M System Architecture

In the system architecture diagram we show the two separate components inside the cloud required for system management and application processing to satisfy privacy laws. Both components may have separate billing options and can run in separate environments. The management station may also include:

- System initialization
- Remote field service options (such as field upgrades, reset to default parameters, and remote test)
- Control for billing purposes (such as account disable, account enable, and billing features)
- Control for theft purposes (the equivalent of bricking the device)

Given this type of architecture, there are additional protocols and programs that should be considered:

- Custom developed management applications on cloud systems

·   SNMP management for collections of sensor nodes

·   Billing integration programs in the cloud

·   Support for discontinuous operation using SQLite running on Unison OS to store and selectively update data to the cloud

Billing is a critical aspect of commercial systems. Telecoms operators have demonstrated that the monthly pay model is the best revenue choice. In addition, automatic service selection and

hardware building blocks called [d.tools], which made it possible to create working digitally connected physical prototypes more easily.

·

In a study in collaboration with 13 IoT startups from the ThingsCon network across 3 major European cities—London, Amsterdam and Berlin—we set up sessions with 4 to 5 startups in each city. During these sessions, the prime focus was on creating a visualized timeline of the design process linked to each startup's product. Afterwards, all gathered processes were analyzed using the 'Development Oriented Triangulation' framework (DOT), which focuses on breaking down design processes into 5 different categories:

·   *Library*: A 'static' research activity, such as looking up existing work or analyzing existing data.

·   *Field*: Going out into the field to observe, trial, or explore something in content.

·   *Showroom*: Creating something, usually a prototype that can be shown and evaluated by experts.

·   *Workshop*: Making something through prototyping or exploring a topic in a 'hands-on'

way.

·   *Lab*: Testing a prototype or idea in a controlled way, not necessarily in a realistic context of use.

Each of the 13 design processes we gathered this way was broken down using that system. During the analysis, a clear structure emerged: All design processes involved 'technical', 'business', and 'user' related design activities.

·   *Technical* activities relate to, for instance, choosing a technology or developing hardware.

·   A *business* activity within the startup context is often related to showcasing a working prototype to investors.

· Lastly, a *user* activity is anything that touches on users' involvement, like a trial session,

or going out into the field to interview relevant people.

Model

The development projects for even the simplest IoT products must integrate technical elements and effectively coordinate the activities of experts from many disciplines across multiple organizations.

There are also new challenges posed by system testing, security, maintenance, support, warranty, regulatory compliance, data governance, and user privacy—IoT development faces unique obstacles in all these areas. No wonder the risks of delays, cost overruns, and overall failures are so great.

To help companies understand the complexity of IoT device development, IoT technology stack model can be referenced below figure 5.2. It shows IoT devices are made up of multiple technology layers including, the physical hardware device, IoT cloud platform, embedded software, network communication protocols, and security.

9. Build a Robot Arm

10. Create a Fully Functional Computer Control Panel

IOT Case studies: Smart City Street lights control & monitoring.

Introduction-Automate street lights are necessary while we are trying to survive in the era of smart world. Automation provides perfection and efficiency. The goal is to automate street lighting, as current system is facing many problems. A user has to deal with numerous problems like maintenance problem, timer problem, connectivity problem, display problem. The solution to these problems is IoT Based Street Lights.

"treet lights are one of the main city's assets which provide safe roads, inviting public areas, and enhanced security in homes, businesses, and city centers. As they use in average 40% of a city's electricity spending which leads to power consumption.

Following are the issues of existing electric system. Connectivity issue-In existing system, connections of street light are done manually. As each connection requires different contractors and if any one of them is not available then it will leads to functionality problem of street lights.

1. Timer Problem-Contractors needs to manage timer settings manually. As timer requires twelve hour of continuous electricity supply, and if in case it is not available, it will delay further timer settings.

2. Maintenance problem-If any of the streets light gets failed or any problem occurs, it's not

resolved immediately.

3. Incorrect Readings-Sometimes exact readings are not shown on to the display. So we cannot conclude how much energy is being consumed which give rise in high billing.

"treetlights are among a city's strategic assets providing safe roads, inviting public areas, and enhanced security in homes, businesses, and city centers. However, they are usually very costly to operate, and they use in average 40% of a city's electricity spending.

This project describes a new economical solution of street light control systems. The control system consists of wireless technology. Base server can control the whole city's street lights by just sending a notification using network.

Component descriptions

1. Energy efficiency using SSL-SSL is nothing but the smart street light system. The SSL system, a framework for fast, reliable, and power efficient street lamp switching based on pedestrians' location and personal desires of safety . In the developed prototype user location, detection as well as safety zone definition and announcement of other configuration information is accomplished using standard Smartphone capabilities. An application on the phone is periodically sending location and other information to the SSL server. For street lamp control, each and every lamppost is extended with a ZigBee-based radio device, receiving control information from the SSL server via multi-hop routing.

2. Embedded Platform for IoT applications- For embedded platforms, CoAP (Constraint Application protocol) is used for IOT applications. The main idea of this protocol is to provide a lightweight protocol for resource-oriented applications run on constrained networks. For

Unit -5

Topics to be covered

IOT Design methodology: Specification -Requirement, process, model, service, functional & operational view.IOT Privacy and security solutions, Raspberry Pi & arduino devices. IOT Case studies: smart city streetlights control & monitoring.

IoT Design methodology: Specification

It has two main perspectives:

1. **IoT Strategy Execution-**This perspective looks at IoT strategy from an enterprise perspective, including IoT strategy definition, IoT opportunity identification, IoT business case and IoT program management.

2. **IoT Solution Delivery-**This perspective looks at then individual IoT solution and the related project. Note that it defines the interfaces to the related asset and its organization, but usually excludes design and manufacturing of the asset itself.

Following factors can also be considered while designing IoT product:

The IoT Expansion Card form factor is designed to be a cost-competitive solution for several current and emerging market demands for host applications, including:

• On-demand hardware applications — The same host application can be used for different market segments by including unique expansion card solutions for each segment. For example, a Wi-Fi / Bluetooth solution for one segment and an environmental sensor for another segment.

• Electrical and feature compatibility across IoT technologies —Availability of several data interfaces allows support for various IoT solutions. For example, digital audio over PCM, application control and data transfer over USB, etc.

• Enables Configure/Built to Order

• Accommodates various PAN technologies, sensors, and other IoT applications

• Using the IoT Expansion Card's modular design, all actors of the M2M value chain can benefit.

• Host applications can, without requiring redesign, inherit new features and interfaces for their products

• Technology specialists such as PAN, LPRF, Industrial Field buses or sensors can bring their expansion cards to market

• System integrators and end customers can easily combine host applications and expansion cards to fit their specialized needs

Requirement

Privacy is an essential implementation requirement. Supported by privacy laws, almost all systems require secure communication to the cloud to ensure personal data cannot be accessed or modified and liabilities are eliminated. Furthermore, the management of devices and the data that appears in the cloud need to be managed separately. Without this feature, users' critical personal information is not protected properly and available to anyone with management access.

Service:

For sensor-driven IoT networks, suite of Internet of Things design services ranging from IoT architecture design can be referenced as:

1. Smart Sensor Design

The smart sensor, or "edge device," sits on the outer edge of the IoT network and collects the

granular data required by businesses, individuals, and intelligent systems. Our services include:

- Development of custom single- or multi-function sensors
- Sensor integration (e.g. acoustic, motion, temperature, moisture)
- Smart power management (e.g. ultra-low-power states)
- Video security and data capture
- Wired or wireless networking enablement and integration

2. IoT Gateway Design

The gateway is the bridge between the sensor network and the cloud. It collects all the data transmitted by the sensors and can also push periodic firmware updates to them.

- Embedded software development of sensor data collection solutions
- Latency management to deliver near real-time performance
- Data flow management for large data sets
- PCB design and I/O integration (e.g. Ethernet, USB)
- Power management
- Cloud connectivity (e.g. cellular, Ethernet, Wi-Fi)

3. Sensor Networking

A business-critical requirement for any IoT network is a secure and highly reliable connectivity infrastructure. Our services include:

- Communications firmware development and cross-platform integration
- Secure communications encryption and device identification
- Support for wireless protocols (e.g. Bluetooth 2.1 and Bluetooth Smart, Wi-Fi, 802.xx.x, Sub-1GHz, 2.4 GHz, custom radio protocols)

·   RF Design

o   Co-location testing

o   Certifications such as FCC part 15, ETSI, EN60251, and others

o   RF tuning and optimization for discrete solutions

o   4GHz Industrial, Scientific, and Medical (ISM) bands

o   868MHz/915MHz – custom point to point and broadcast protocols for sensor data, status messages, distributing firmware upgrades and media content

4. Cloud Connectivity

Many types of IoT data translate into true business value via connectivity with the cloud. Have the gateway send periodic "heartbeats" to the cloud server to let it know that the sensor network is alive and healthy

·   Have the gateway and smart sensor devices receive "pushed" commands, configuration

and software updates

·   Support for application-level system management and analytics software

Functional & Operational view

Following table summarizes functional and operational view as:

integration for seamless billing is important. Also credit card dependence creates issues including over the limit issues, expired cards and deleted accounts.

Self-supporting users are a key to implementation success, too. This includes things like remote field service so devices never return to the factory, intelligent or automatic configuration, online help, community help, and very intuitive products are all key.

Application integration is also important. Today point systems predominate, but in the future the key will be making sensors available to a broad set of applications that the user chooses. Accuracy and reliability can substantially influence results application results and competition is expected in this area as soon as standard interfaces emerge. Indirect access via a server ensures security, evolution without application changes and billing control.

Discontinuous Operation and Big Data go hand in hand. With devices connecting and disconnecting randomly, a need to preserve data for the sensors and update the cloud later is required. Storage limitations exist for both power and cost reasons. If some data is critical, it may be saved while other data is discarded. All data might be saved and a selective update to the cloud performed later. Algorithms to process the data can run in either the cloud or the sensors or any intermediate nodes. All of these options present particular challenges to the sensor, cloud, communications, and external applications.

Multiple connection sensor access is also a requirement to make sensors truly available to a broad set of applications. This connection will most likely happen through a server to simplify the sensors and eliminate power requirements for duplicate messages.

Process:

Three types of actions have been extensively recognized and
researched in IoT design Process:

· *In-context artefacts*: By placing working products or prototypes in context, we can obtain insights into how people live with networked products in their daily lives. This allows us to understand in which ways technology can be used to support people in their day-to-day efforts, or which values we could focus on. A good example of this is Bill Gaver's history tablecloth, a digitally connected product created as an experiential prototype. Only by living with this product on a day-to-day basis you can fully experience and understand its functionality and value.

· *Involving stakeholders*: When it comes to designing with 'novel' technologies, it is not easy to present ideas to end users or other stakeholders in early stages of the design process. But the literature thoroughly explores way to better explore concepts and early ideas. A school book example of such a technique is Wizard of Oz testing, which originated in the 1980s. Wizard of Oz testing comes down to providing an end user with the illusion that a digital system is functional, whereas it is actually being controlled manually.

· *Design tools*: When designing digitally connected physical products, a design challenge is to constantly keep track of which design choices in either medium influence each other. For example, adding a physical button to a hardware device might be impossible because the micro controller does not have any spare pins to use: In this case, a digital constraint limits a physical design choice. Research by Bjorn Hartmann (2006) and others has been tackling these challenges. Hartmann created a set of software and

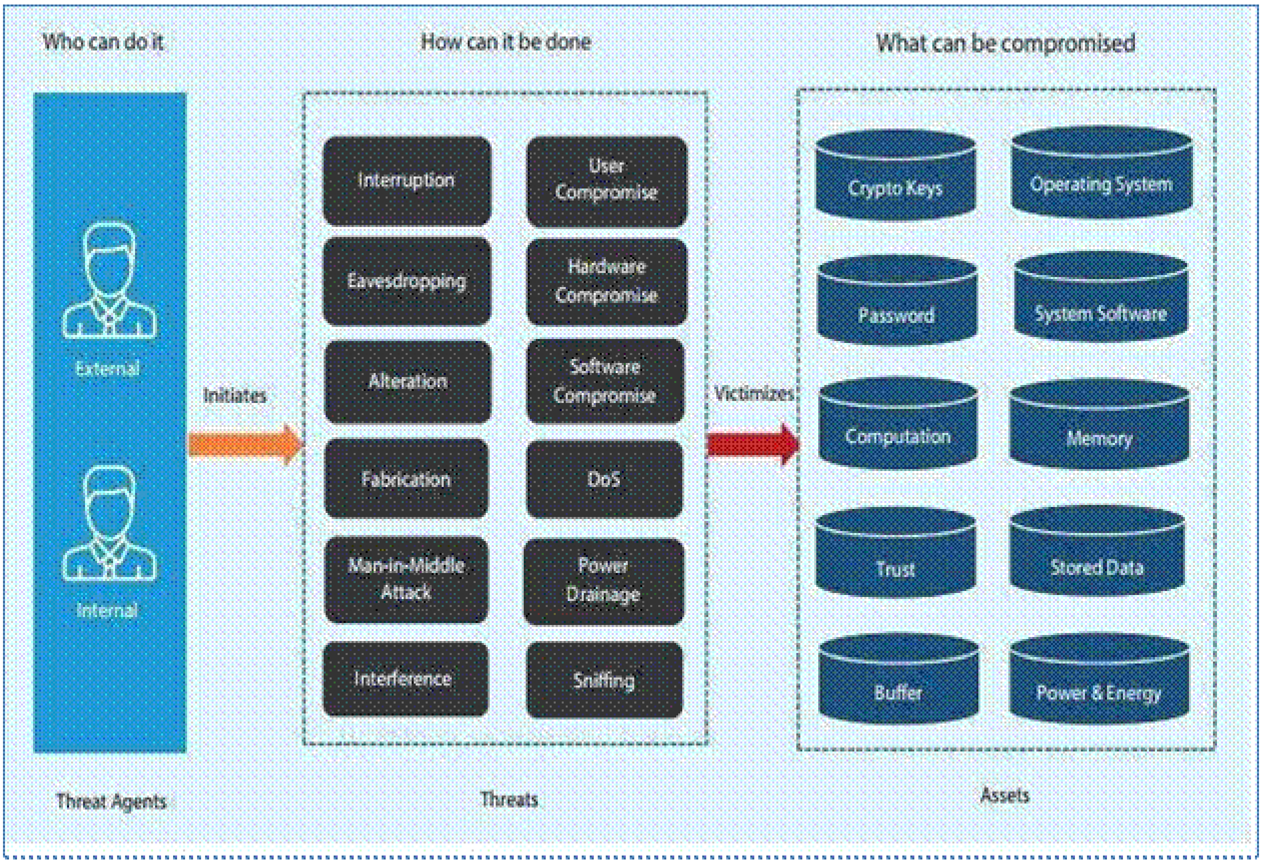

Fig .5.3 Privacy and security Architecture

A major difference between traditional Internet and the IoT is the amount of data being collected about the user. Data is collected universally in the IoT and this data can be used to build an invasive profile of the consumer. The organizations recognized three major privacy concerns: facilitation of the collection of large amounts of consumer data, using that data in ways unexpected by the consumer, and security of data. This ubiquitous data collection makes the Internet of Things a much more data driven economy. With massive quantities of continuous data, new discoveries can be made, but little to no regulation can by harmful to the consumers. Privacy issues are especially hard to discuss because, by nature, privacy is subjective. The organizations aim to promote three best practices: privacy by design, simplified consumer choice, transparency. Companies have to make an effort to build consumer protection in from the beginning.

With such an asymmetry of power between businesses and their consumers, the organizations are looking for ways to protect users against abuse of their data. The IoT, a data-driven ecosystem, requires a trust between the business and consumer that exists even now. A user shares data with a business and in return receives a service. The organizations is seeking to push businesses and companies towards built-in security and designing security into new devices. For the IoT, the data is usually passively and ubiquitously collected. As a result, the organizations believes businesses will have to earn user trust and at a data level, which means involving the user. A similar problem exists in the energy industry. A Green Button was created reducing the burdens of manufacturers, we have designed our software framework for embedded system nodes to allow IoT service development with minimal efforts. As this framework supports application-layer API, which does not affect the existing codes and hides network-layer functions, product manufacturers only need to append a simple CoAP service definition, network driver, and physical network adapter to start IoT services on nodes.

3. **Electrical power saving using VANET-**The huge amount of electrical power of many countries is consumed in lighting the streets. However, vehicles pass with very low rate in specific periods of time and parts of the streets are not occupied by vehicles over time. An efficient autonomous street lighting control and monitoring system based on the innovative technology named as Vehicular Ad-Hoc Networks (VANET) is proposed. The system can be integrated with VANET to reduce the cost and use the rich services and communication features of VANET. Huge energy can be saved without affecting the visibility and the safety of the drivers. It can extend the lifetime of the lamps. It can automatically monitor the street lighting equipment's and warn the maintenance traffic authority upon failure detection in any place of the streets.

4. **Fully controlled street lights using Raspberry-pi and Zigbee-**The Raspberry-Pi has been chosen for its low costs and for the possibility to drive also a WiMAX modem/router which allows to make the data system visible by a web site accessible by Internet also for areas very far from the city and not reached neither by the ADSL line nor by 3G signals. Intelligent lighting of the lamp, the storage of the functioning data, and their sharing by a local communication wireless mesh realized by ZigBee devices that send information to the coordinator lamp equipped with a RaspberryPi card.

5. **System-**Raspberry-Pi is used to provide interface between user and system. It is connected to wireless network and relay circuit which will pass the operational admin's message to the system. Then relay circuit operate the commands like ON Lights, OFF Lights, Alter ON, Alter OFF onto the connected array of street light. Our system includes two admin: System admin and Operational admin. System admin handles log messages and operational admin. System admin can add, delete and view operational admin. Once the operational admin added to the system by the system admin then operational admin can log in to the system. For example, operational admin choose the city and area from database to ON or OFF the street lights. And if any fault occurs in the functioning of street lights then relay circuit will send the faulty street light's IP address to the operational admin then operation admin will resolve the problem.

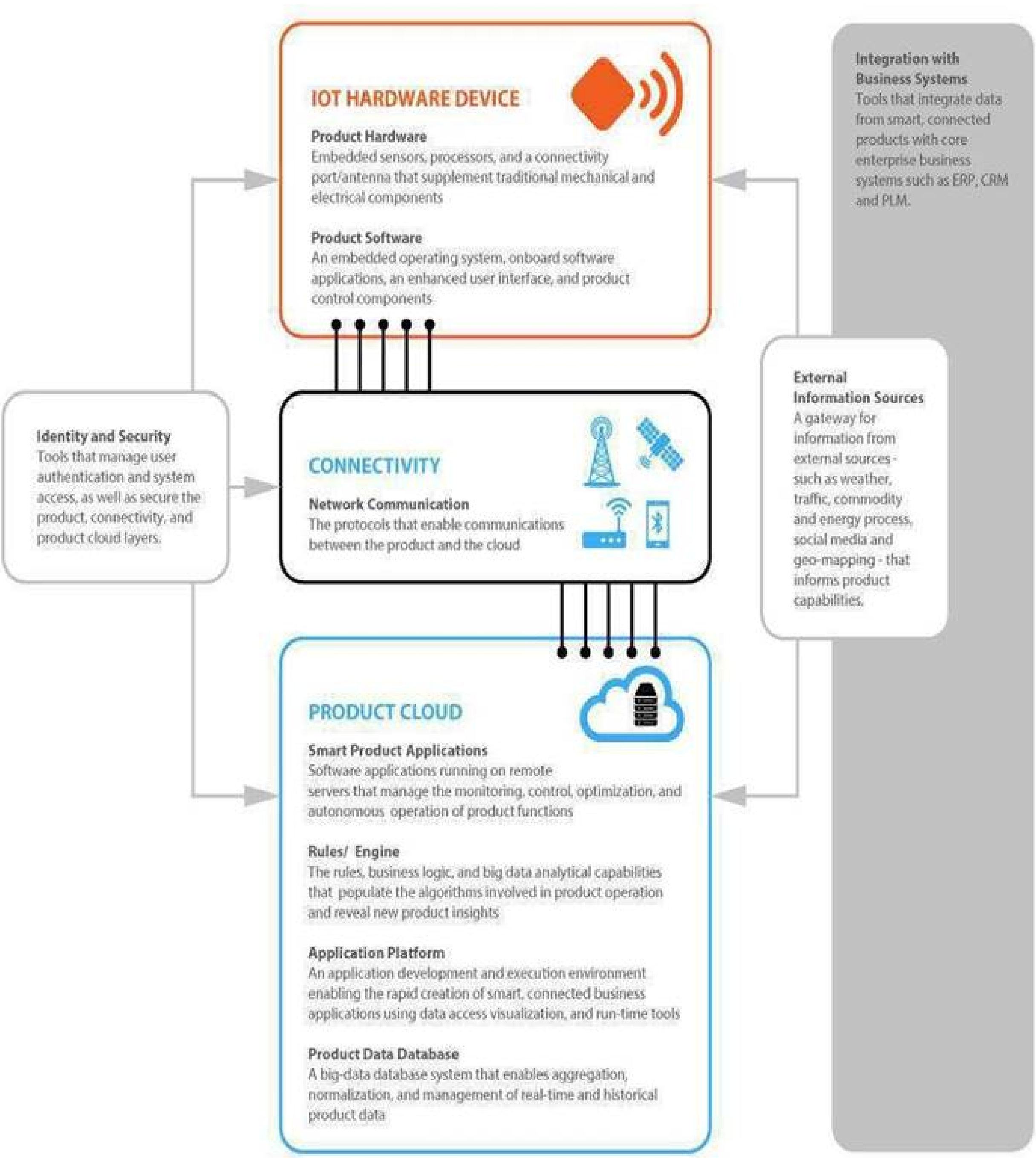

Fig .5.2 IoT Technology Stack Model

Definition	Describes how the system will be operated, administered, and supported when it is running in its production environment

Concerns	· installation and upgrade · functional migration · data migration · operational monitoring and control · alerting · configuration management · performance monitoring · support · backup and restore · operation in third-party environments
Models	· installation models · migration models · configuration management models · administration models · support models
Pitfalls	· lack of engagement with the operational staff · lack of backout planning · lack of migration planning · insufficient migration window · missing management tools · production environment constraints, · lack of integration into the production environment · inadequate backup models · unsuitable alerting
Stakeholders	System administrators, production engineers, developers, testers, communicators, and assessors

Applicability	Any system being deployed into a complex or critical operational environment

IOT Privacy and security solutions-The IoT has to protect against attacks from the following categories: authentication, access control, confidentiality, integrity, and availability. Authentication involves the mutual verification of routing peers before they share route information and ensures shared data origin is accurate. In the IoT, authentication has to be strong and highly automated. Access control is the prevention of unauthorized node use, i.e. making sure nodes are not compromised. Confidentiality is the protection of information, especially when shared over a publicly accessible medium such as air for wireless. Integrity involves the protection of data and confirms no unauthorized modifications occur.

in order to standardize energy usage information, allow the consumers to download the information, and enlighten the users how their data is being used. Empowering and educating the consumer would help facilitate the integration of the IoT into our everyday lives.

Raspberry Pi:

1. Introduction

"Pi is a single-board computer". Pi is a small scale computer in the size little bigger than a credit card, it packs enough power to run games, word processor like open office, image editor like Gimp and any program of similar magnitude.

Pi was introduced as an educational gadget to be used for protyping by hobbyists and for those who want to learn more about programming. It certainly cannot be a substitute for our day to day Linux, Mac or Windows PC.

Pi is based on a Broadcom SoC (System of Chip) with an ARM processor [~700 MHz], a GPU and 256 to 512 MB RAM. The boot media is an SD card [which is not included], and the SD card can also be used for persist data. Now that you know that the RAM and processing power are not nearly close to the power house machines you might have at home, these Pi's can be used as a Cheap computer for some basic functions, especially for experiments and education. The Pi comes in three Configurations and we will discuss the specifications of those in the coming sections. The cost of a Pi is around $35 for a B Model and is available through many online and physical stores.

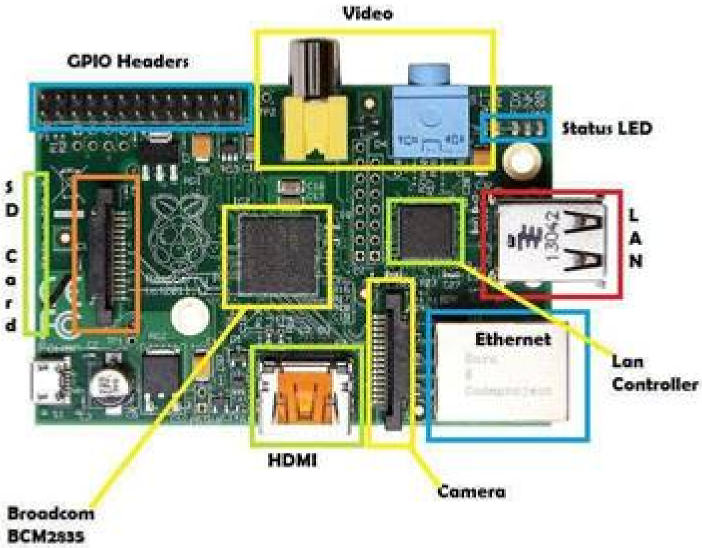

Fig 5.4: Raspberry Pi Board

2. System Specifications

Computer	A Raspberry Pi
Storage	SD Card and a SD card reader to image the OS [These days laptops have inbuilt card readers]
Power supply	5 volt micro USB adapter, mostly your android phone charger would work

Display	An TV/Monitor with DVI or HDMI port
Display connector	HDMI cable or HDMI to DVI converter cable
Input	USB Mouse
Input	USB Keyboard
Network	Ethernet cable
Case	If you really need one, you can get them online based on the model you have

3. Raspberry Pi uses

It can be used in following areas/applications:

1. Retro Gaming
2. Raspberry Pi Tablet
3. Low-Cost Desktop PC
4. Raspberry Pi Cluster
5. Raspberry Pi Cloud Server
6. Raspberry Pi Media Center
7. Web Server
8. Home Automation System
9. VPN
10. Robotics

Arduino devices

1. Introduction

Arduino is an open source computer hardware and software company, project, and user community that designs and manufactures single board microcontrollers and microcontroller kits for building digital devices and interactive objects that can sense and control objects in the physical world. The project's products are distributed as open-source hardware and software, which are licensed under the GNU Lesser General Public License (LGPL) or the GNU General Public License (GPL), permitting the manufacture of Arduino boards and software distribution by anyone. Arduino boards are available commercially in preassembled form, or as do-it-yourself (DIY) kits.

Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The boards feature serial communications interfaces, including Universal Serial Bus (USB) on some models, which are also used for loading programs from personal computers. The microcontrollers are typically programmed using a dialect of features from the programming languages C and C++. In addition to using traditional compiler tool chains, the Arduino project provides an integrated development environment (IDE) based on the Processing language project.

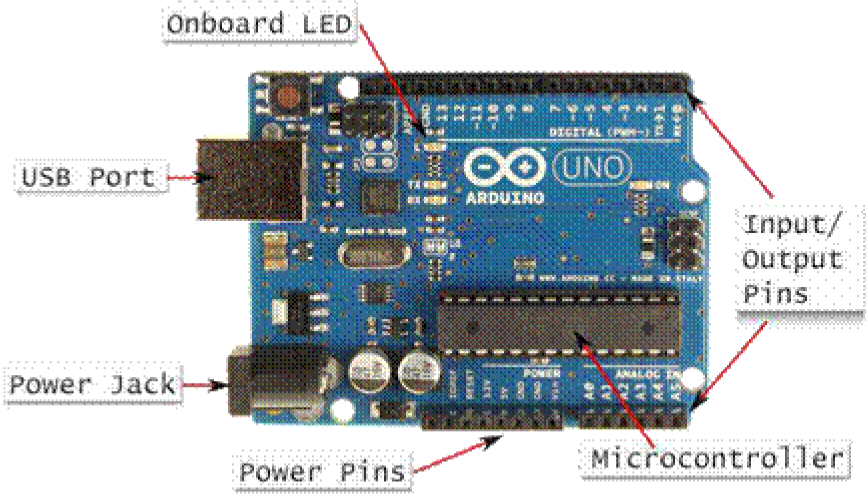

Fig 5.5: Raspberry Pi Boar

2. System Specifications

Component	Specification
Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328)
SRAM	2 KB (ATmega328)

EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz
Length	68.6 mm
Width	53.4 mm
Weight	25 g

3. Arduino uses

It can be used in following areas/applications:

1. Tiny Weather display
2. MIDI Controller
3. Fingerprint Scanner to Your Garage Door Opener
4. Auto-Trigger Spray Gun
5. Make Your Own Arduino
6. Add Motion-Triggered Night Lights Under the Bed
7. Mute Any Phrase You Want on Your TV
8. Add an Ambilight Sensor to Your LCD Display

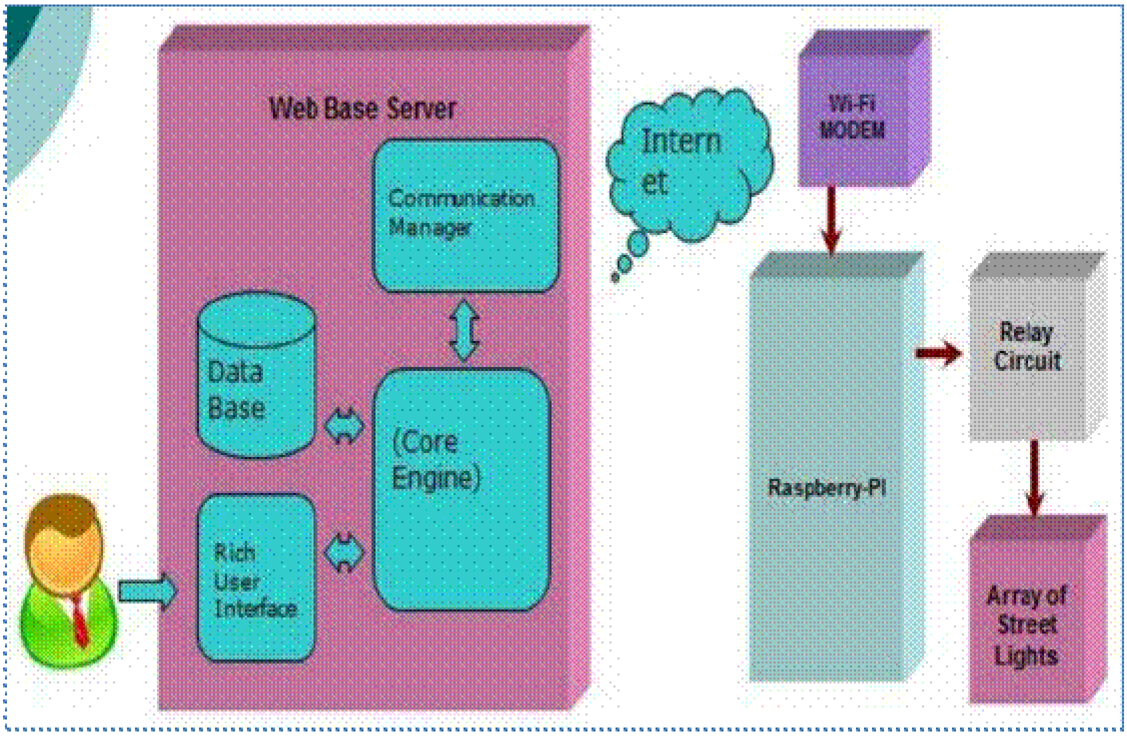

Fig 5.6: System architecture of system