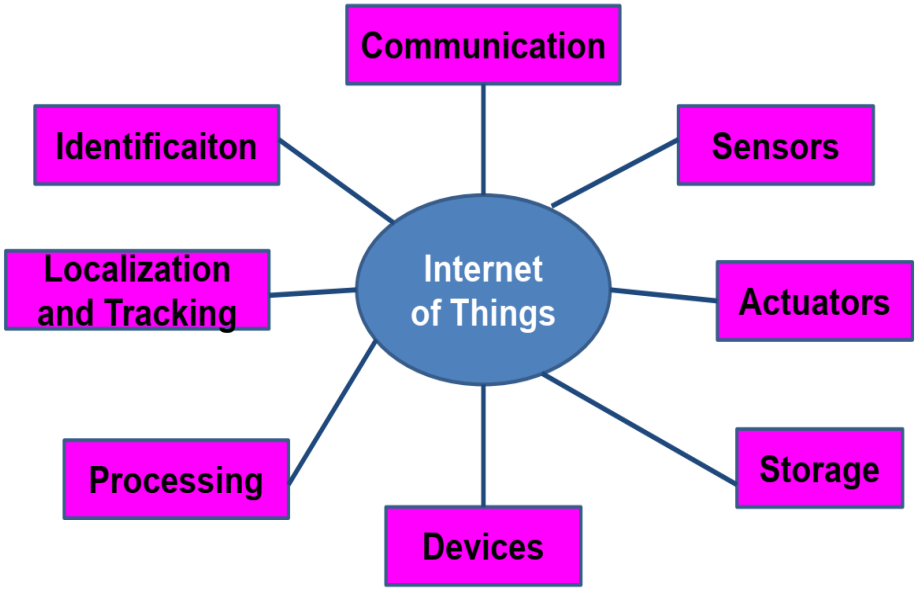
Chapter 2

IoT Architecture

IoT is consists of the following components



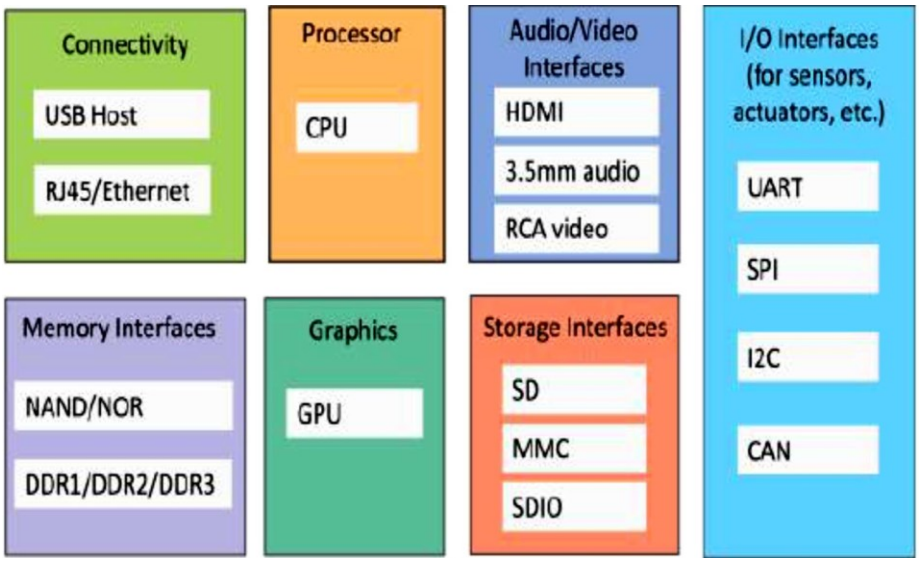
IoT Components

- Things we connect : Hardware, sensors, actuators
- Connectivity : Medium we use to connect things
- Platfrom : processing and storing collected data
  - Receive and send data via standardized interfaces or API
  - Store the data
  - Process data
- Analytics
  - Get insights from gathered data
  - User interface

IoT Devices

- The "Things" in IoT usually refers to IoT devices which have unique identities
- They can perform remote sensing, actuating and monitoring capabilities
- IoT devices can:
  - Exchange data with other connected devices and applications
  - Collect data from other devices and process the data locally
  - Send the data to centralized servers or cloudbased application back-ends for processing
  - Perform some tasks locally and other tasks within the IoT infrastructure

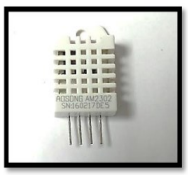
Generic Block Diagram of IoT Device



SMART OBJECTS: THE "THINGS" IN IoT

- Sensors
  - Characteristic of any device or material to detect presence of particular physical quantity
  - Output of sensor is signal, which is converted to human readable form
  - Performs some function of input by sensing or feeling physical changes in the characteristic of a system in response to stimuli
  - Input: Physical parameter or stimuli
    - Example: Temperature, light, gas, pressure, and sound
  - Output: Response to stimuli

Sensors

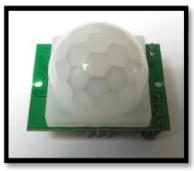


Temperature and Humidity sensor – DH22

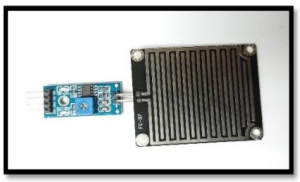
Gas (LPG, CH4, and CO) detector sensor - MQ-5

Ultrasonic sensor - HC-SR04

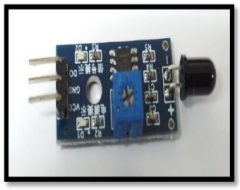
CMOS Camera



PIR Sensor



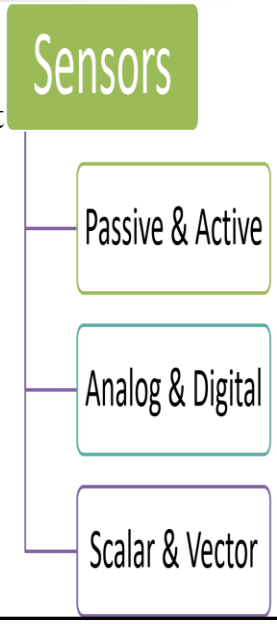
Rain Detector Sensor



Fire Detector Sensor

Classification

- Passive Sensor
  - Cannot independently sense the input
  - Example:
    - Accelerometer
    - Soil Moisture
    - Water-level and
    - Temperature Sensors
- Active Sensor
  - Independently sense the input
  - Example:
    - Radar
    - Sounder and
    - Laser Altimeter Sensors

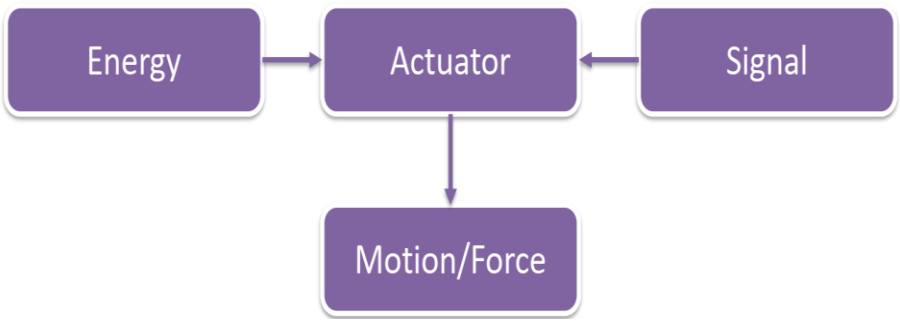


- Analog Sensor
  - The response or output of the sensor is some continuous function of its input parameter
  - Example:
    - Temperature sensor
    - LDR
    - Analog Pressure Sensor and
    - Analog Hall Effect/Magnetic Sensor
- Digital Sensor
  - Responses in binary nature
  - Designs to overcome the disadvantages of analog sensors
  - Along with the analog sensor it also comprises of extra electronics for bit conversion
  - Example:
    - Passive Infrared (PIR) Sensor and
    - Digital Temperature Sensor (DS1620)

- Scalar Sensor
  - Detects the input parameter only based on its magnitude
  - Response of the sensor is a function of magnitude of the input parameter
  - Not affected by direction of input parameter
  - Example: Temperature, Gas, Strain, Color and Smoke Sensors
- Vector Sensor
  - Response of sensor depends on magnitude of direction and orientation of input parameter
  - Example :
    - Accelerometer, Gyroscope, Magnetic Field
    - Motion Detector Sensors

Actuator

- Part of system that deals with the control action required (mechanical action)
- Mechanical or electro-mechanical devices



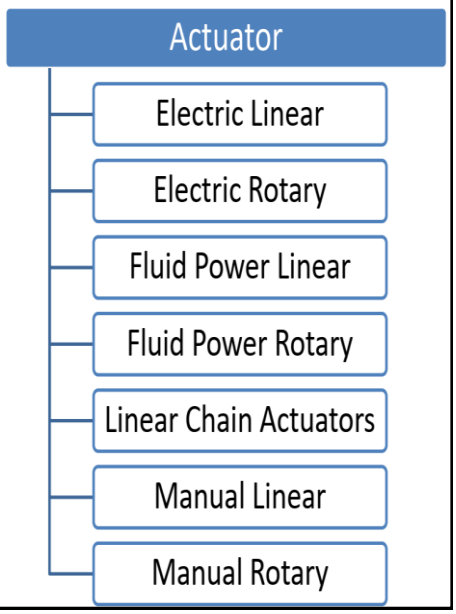
Actuator

- A control signal is input to an actuator and an energy source is necessary for its operation
- Available in both micro and macro scales
- Example:
  - Electric Motor,
  - Solenoid,
  - Hard Drive Stepper Motor
  - Comb Drive
  - Hydraulic Cylinder,
  - Piezoelectric Actuator
  - Pneumatic Actuator

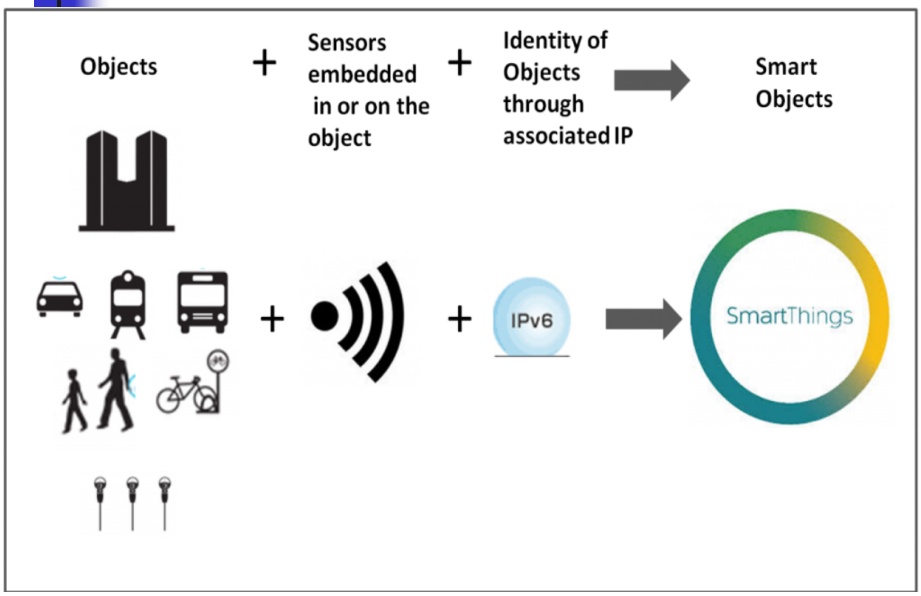
Actuator - Classification

DC Motor

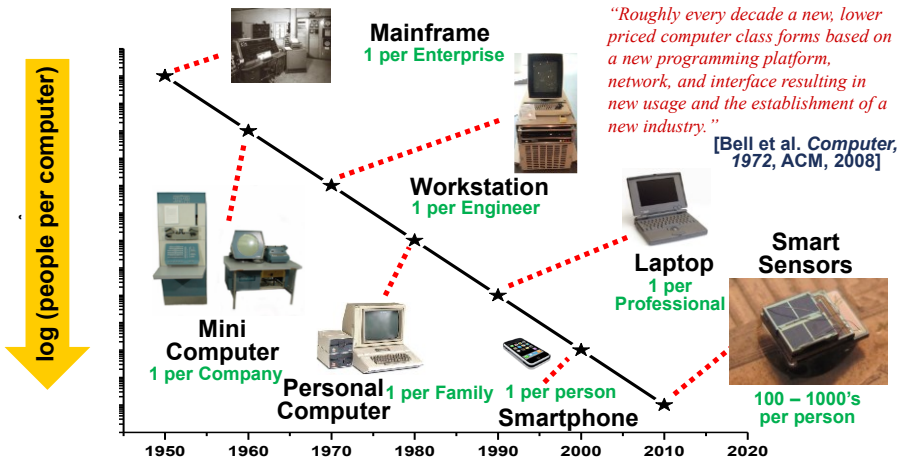
Relay



Smart Objects



Bell's Law of Computer Classes: A new computer class emerges roughly every decade

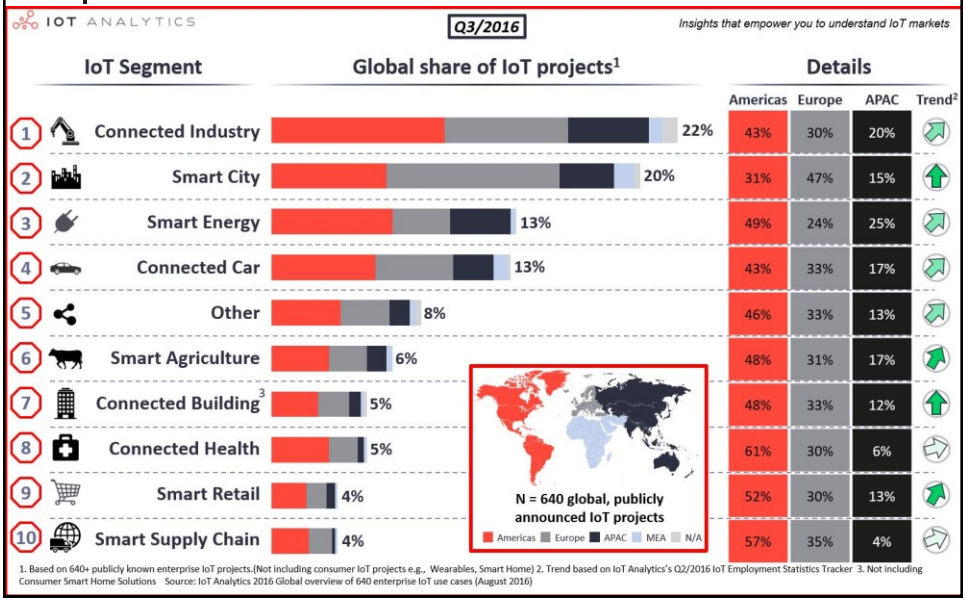


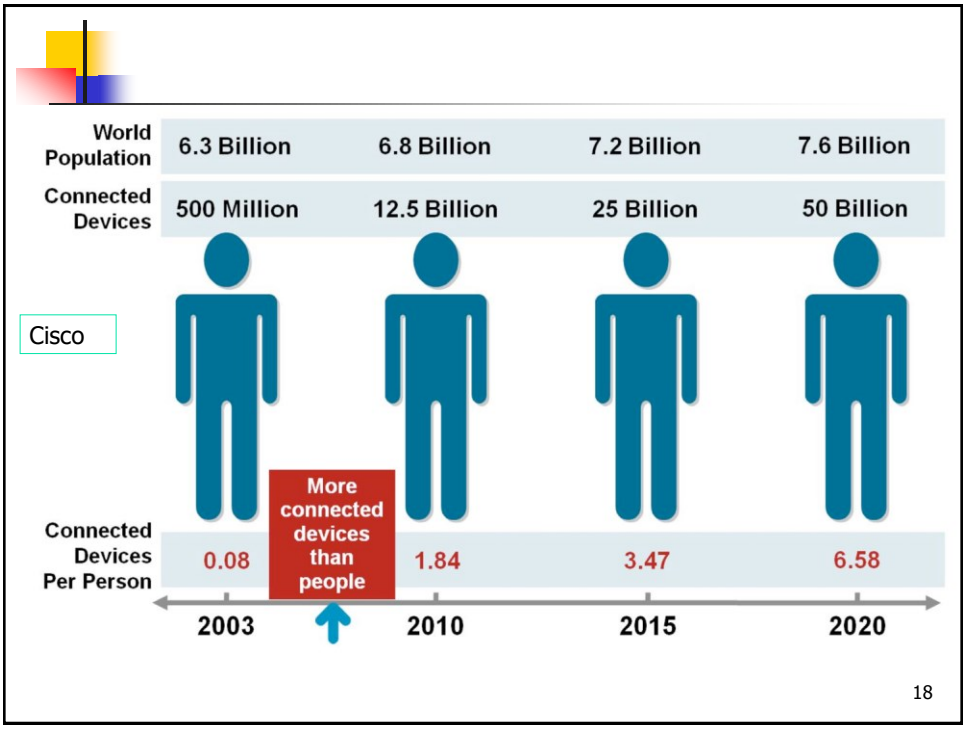
"Roughly every decade a new, lower priced computer class forms based on a new programming platform, network, and interface resulting in new usage and the establishment of a new industry."

[Bell et al. *Computer, 1972*, ACM, 2008]

16

IoT analytics

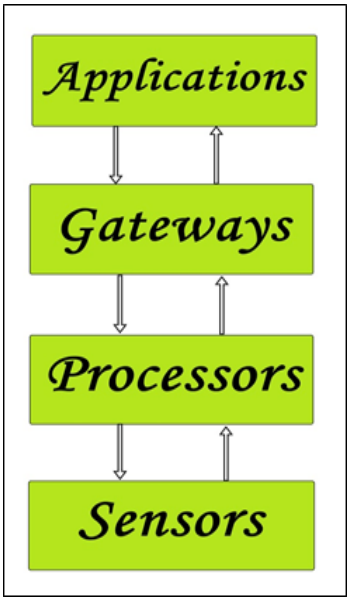




18

BUILDING BLOCKS of IoT

- Four things form basic building blocks of the IoT system –sensors, processors, gateways, applications. Each of these nodes has to have its own characteristics in order to form an useful IoT system.



Sensors:

- These form the front end of the IoT devices. These are the so-called "Things" of the system. Their main purpose is to collect data from its surroundings (sensors) or give out data to its surrounding (actuators).
- These have to be uniquely identifiable devices with a unique IP address so that they can be easily identifiable over a large network.
- These have to be active in nature which means that they should be able to collect real-time data. These can either work on their own (autonomous in nature) or can be made to work by the user depending on their needs (user-controlled).
- Examples of sensors are gas sensor, water quality sensor, moisture sensor, etc

Processors:

- Processors are the brain of the IoT system. Their main function is to process the data captured by the sensors and process them so as to extract the valuable data from the enormous amount of raw data collected. In a word, we can say that it gives intelligence to the data.
- Processors mostly work on real-time basis and can be easily controlled by applications. These are also responsible for securing the data – that is performing encryption and decryption of data.
- Embedded hardware devices, microcontroller, etc are the ones that process the data because they have processors attached to it.

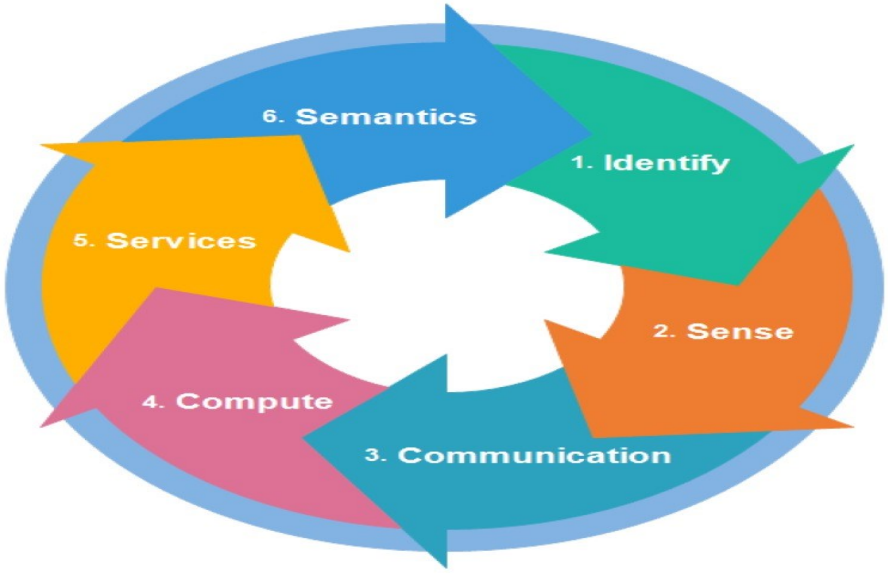
Gateways:

- Gateways are responsible for routing the processed data and send it to proper locations for its (data) proper utilization.
- In other words, we can say that gateway helps in to and fro communication of the data. It provides network connectivity to the data. Network connectivity is essential for any IoT system to communicate.
- LAN, WAN, PAN, etc are examples of network gateways.

Applications:

- Applications form another end of an IoT system. Applications are essential for proper utilization of all the data collected.
- These cloud-based applications which are responsible for rendering the effective meaning to the data collected. Applications are controlled by users and are a delivery point of particular services.
- Examples of applications are home automation apps, security systems, industrial control hub, etc.

Elements of IoT



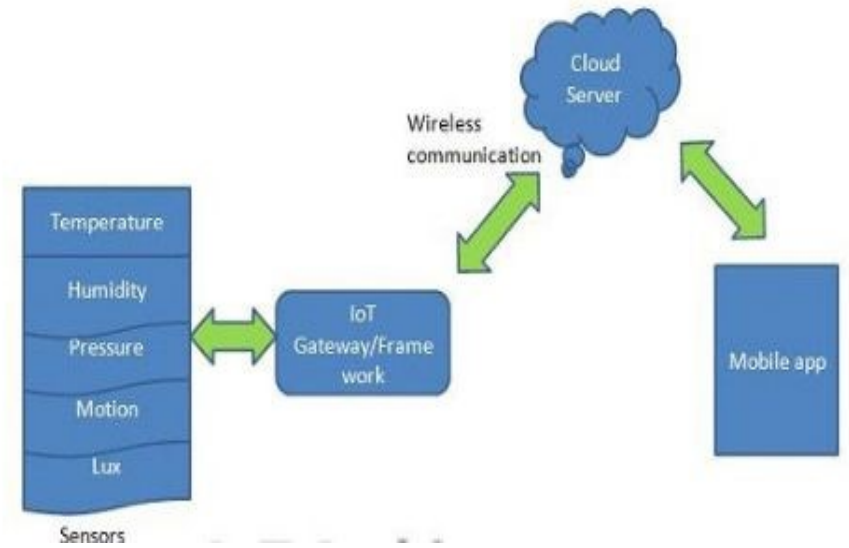
The elements and key technologies of IoT.

IoT Elements		Technologies
Identification	Naming	Electronic, Product Code, Ucode
	Addressing	IPv4, and IPv6
Sensing		Smart, Sensors, RFID Tags, Wearable Sensing Devices and Actuators
Communication		Radio Frequency Identification, Wireless Sensor Network, Near Field Communication (NFC), Bluetooth, Long Term Evolution (LTE)
Computation	Hardware	Audrino, Raspherry Pi, Intel Galil
	Software	Operating System
Services		Identity-Related, Information Aggregation, Collaborative-Aware and Ubiquitous
Semantics		RDF, OWL, EXI

The IoT Architectural landscape

- Thousands of new applications spanning numerous domains.
- Each comes with its own requirements; combining these leads to complex (difficult to manage) and often proprietary systems.
- Defining a unified architecture is challenging and interoperability problematic if too many standards to chose from
- Documentation scattered and often difficult to navigate.

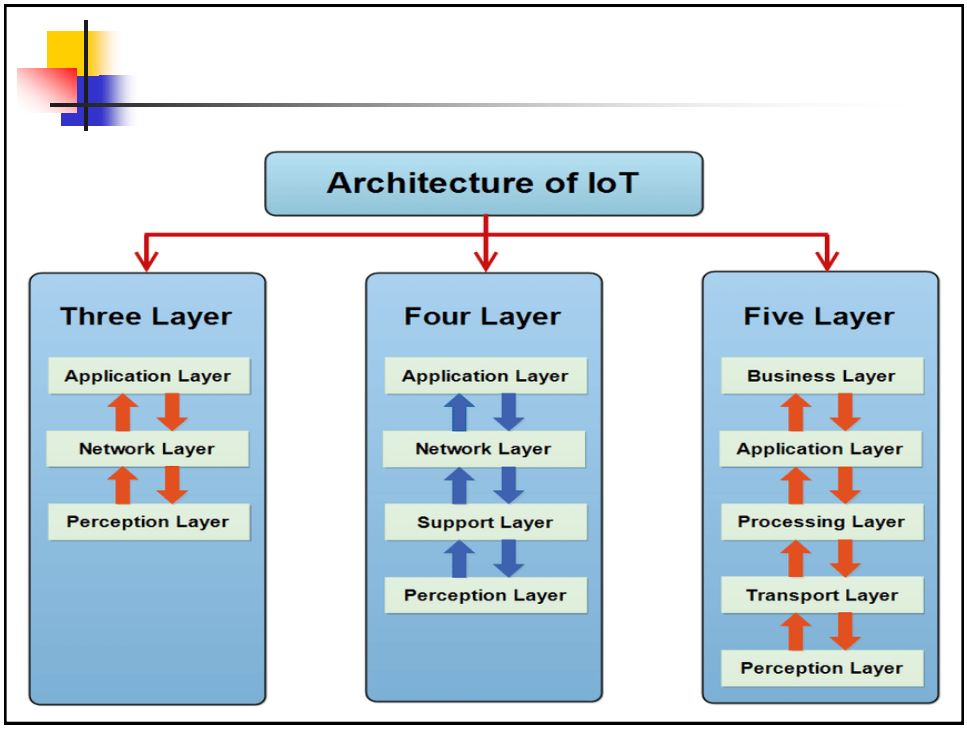
IoT Architecture



- An **IoT architecture** consists of your networked things, typically wireless sensors and actuators.
- It includes sensor data aggregation systems and analog-to-digital data conversion
- It includes the processing and business .The five layers are perception, transport, processing, application, and business layers.

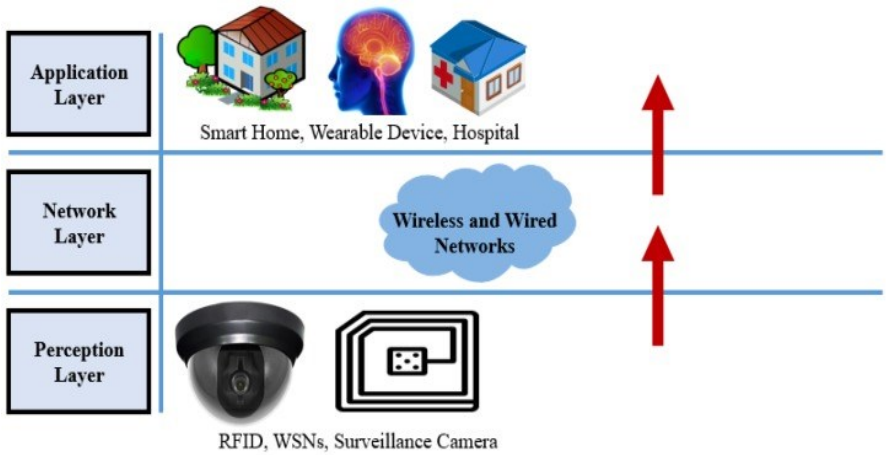
How do IoT Work?

- An **IoT system** consists of sensors/devices which "talk" to the cloud through some kind of connectivity. Once the data gets to the cloud, software processes it and then might decide to perform an action, such as sending an alert or automatically adjusting the sensors/devices without the need for the user



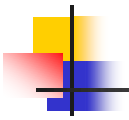
The three-layered architecture of IoT.

- It is a very basic architecture and fulfills the basic idea of IoT



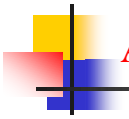
Perception Layer (Lớp cảm nhận)

- It is also known as a **sensor layer**.
- It works like people's eyes, ears and nose.
- It has the responsibility to identify things and collect the information from them.
- There are many types of sensors attached to objects to collect information such as RFID, 2-D barcode and sensors.
- The sensors are chosen according to the requirement of applications.
- The information that is collected by these sensors can be about location, changes in the air, environment, motion, vibration, etc.



Network Layer

- Network layer is also known as transmission layer.
- It acts like a bridge between perception layer and application layer.
- It carries and transmits the information collected from the physical objects through sensors.
- The medium for the transmission can be wireless or wire based.
- It also takes the responsibility for connecting the smart things, network devices and networks to each other.

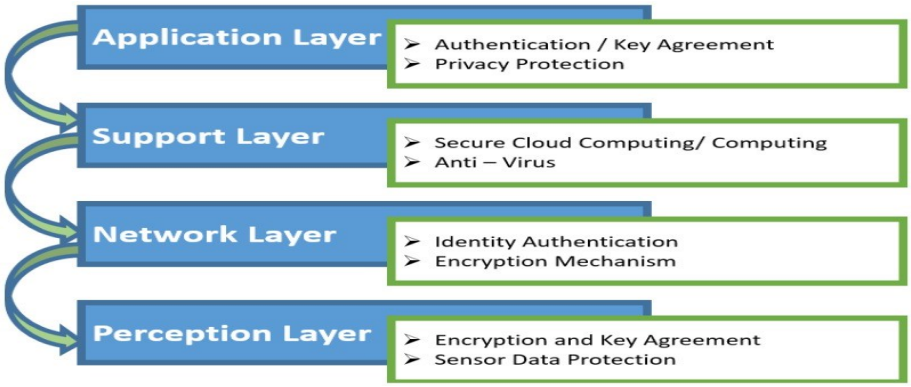


Application Layer

- Application layer defines all applications that use the IoT technology or in which IoT has deployed.
- The applications of IoT can be smart homes, smart cities, smart health, animal tracking, etc.
- It has the responsibility to provide the services to the applications. The services may be varying for each application because services depend on the information that is collected by sensors.

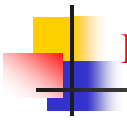
Four Layer Architecture

- The three-layer architecture was most basic architecture. Due to continuous development in IoT, it could not fulfill all the requirements of IoT. Therefore, researchers proposed an architecture with four layers



Support Layer

- The reason to make a fourth layer is the security in architecture of IoT
- Information is sent directly to the network layer in three-layer architecture.
- Due to sending information directly to the network layer, the chances of getting threats increase. Due to flaws that were available in three-layer architecture, a new layer is proposed.
- In four-layer architecture, information is sent to a support layer that is obtained from a perception layer. The support layer has two responsibilities.
  - It confirms that information is sent by the authentic users and protected from threats



Five Layer Architecture

- The four-layer architecture played an important role in the development of IoT.
- There were also some issues regarding security and storage in four-layer architecture. Researchers proposed five-layer architecture to make the IoT secure
- It has three layers like previous architectures whose names are perception layer, transport layer and application layer.
- It also has two more layers. The names of these newly proposed layers are processing layer and business layer

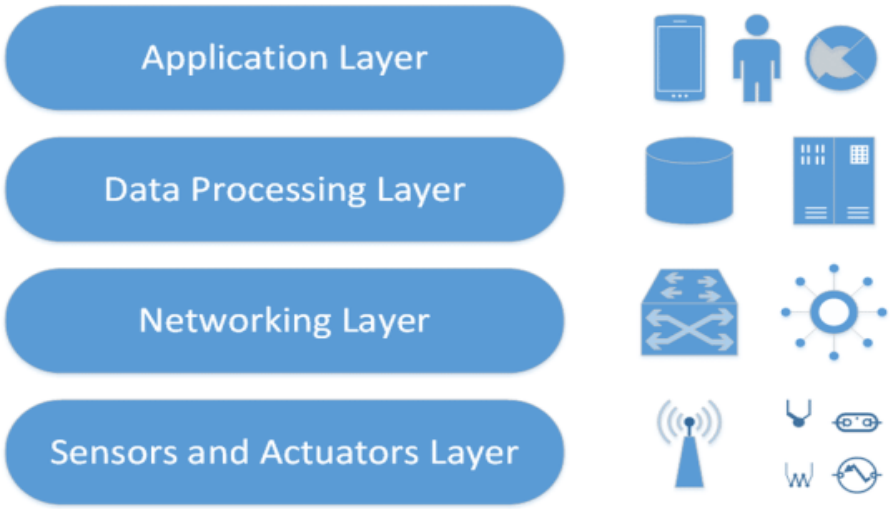
Processing Layer

- The processing layer is also known as a middleware layer
- It collects the information that is sent from a transport layer. It performs processing onto the collected information.
- It has the responsibility to eliminate extra information that has no meaning and extracts the useful information
- However, it also removes the problem of big data in IoT. In big data, a large amount of information is received which can affect performance of IoT. T

Business Layer

- The business layer refers to an intended behavior of an application and acts like a manager of a whole system.
- It has responsibilities to manage and control applications, business and profits models of IoT. The user's privacy is also managed by this layer.
- It also has the ability to determine how information can be created, stored and changed.

More details layer



Application Layer

- The application layer defines all applications in which IoT has deployed
- IoT Applications such as:
  - Smart home
  - Smart health
  - Smart cities
- It has the authority to provide services to the applications.
- The services may be different for each application because of services based on the information collected by sensors.

Data Processing Layer

- In three-layer architecture, the data were directly sent to the network layer
- In four-layer architecture, data is sent to this layer that is obtained from a perception layer.
- Data Processing Layer has two responsibilities it confirms that data is forwarded by the authentic users and prevented from threats.

Network Layer

- This layer is also known as a transmission layer.
- It acts like a bridge that carries and transmits data gathered from physical objects through sensors.
- The medium can be wireless or wire-based.
- It also connects the network devices and networks to each other

Perception layer/Sensor layer

- The sensor layer has the responsibility to recognize things and gather the data from them.
- There are many types of sensors connected to the objects to gather information such as RFID, sensors and 2-D barcode
- The sensors are selected as per the requirement of applications
- The data that is collected by these sensors can be about location, changes in the air, environment, etc.

Architecture

- An architecture (of a system) is
  "*The fundamental organization of a system embodied by its components [jl:building blocks], their relationships to each other [jl: connectors and interfaces,dependencies] and to the environment and the principles guiding its design [jl:rationales for choices, rules & constraints for building blocks and connectors]*
  and evolution"

- An architecture *description* is
  - a collection of *models* organized into *views* that examine a system from a certain *viewpoint* defined by the *concern* of a stakeholder
  - for *understanding, analysis, communication, construction, documentation*

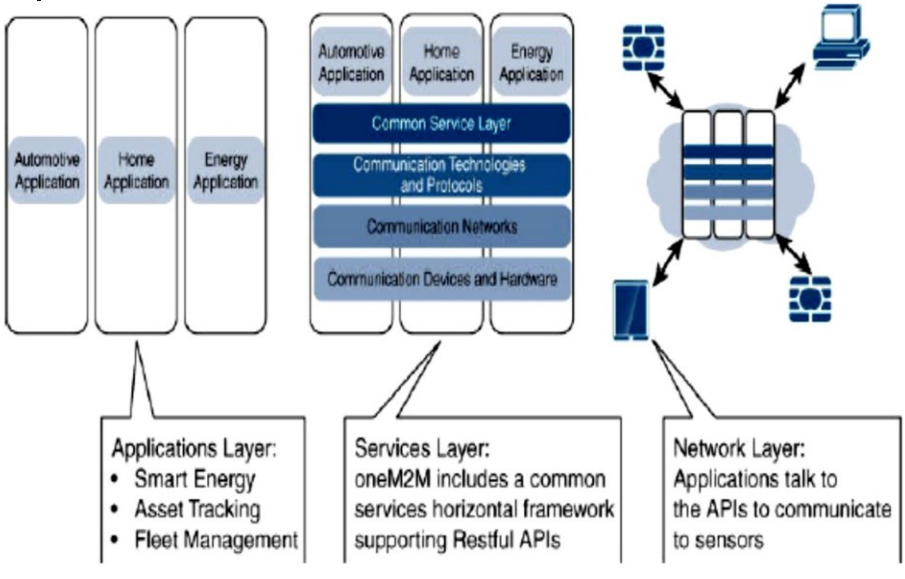
Main design principles

- **Main design principles and needed capabilities**
  - The approach taken in SENSEI was to develop an architecture and technology building blocks that enable a "Real World integration in a future Internet."
  - The telecommunications industry, meanwhile, has focused on defining a common service core for supporting various M2M applications
  - The approach taken in IoT-A differs from the two approaches above in the sense that instead of defining a single architecture, a reference architecture is created, captured in what the IoT-A refers to as the Architectural Reference Model (ARM).

- Design for reuse of deployed IoT resources across application domains
- Design for different abstraction levels that hide underlying complexities and heterogeneities.
- Design for sensing and actors taking on different roles of providing and using services across different business domains and value chains.
- Design for ensuring trust, security, and privacy.
- Design for scalability, performance, and effectiveness.
- Design for evaluability, heterogeneity, and simplicity of integration

- Design for simplicity of management.
- Design for different service delivery models.
- Design for lifecycle support. The lifecycle phases are: planning, development, deployment, and execution. Management aspects include deployment efficiency, design time tools, and run-time management.

One M2M IoT Standardized Architecture

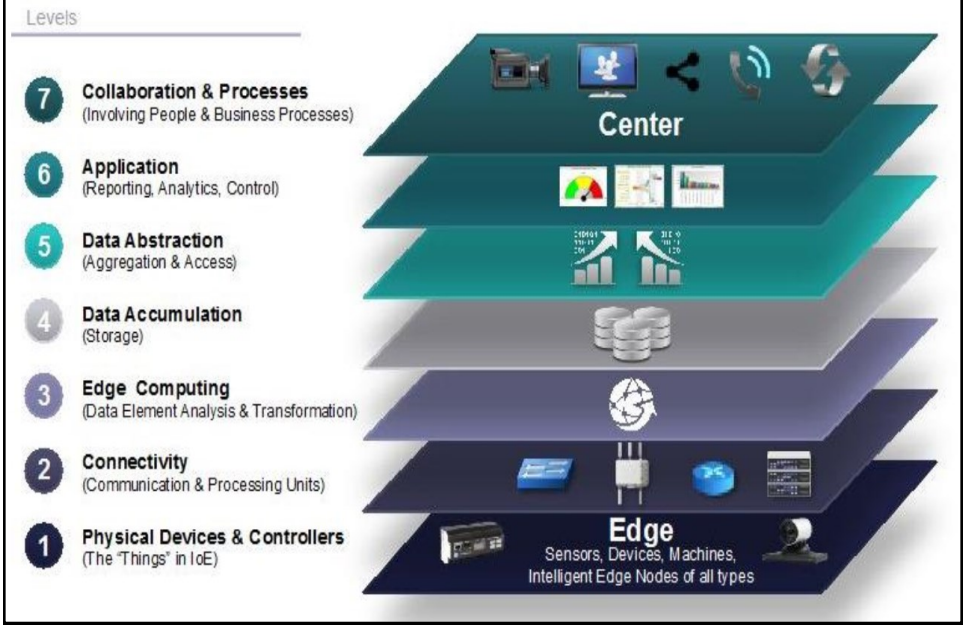


- The one M2M architecture divides IoT functions into three major domains:
  - Application Layer
  - Services Layer
  - Network Layer
- Application Layer
  - The one M2M architecture gives major focus on connectivity between devices and their applications
  - Includes the application-layer protocols
  - Attempts to standardize northbound API definitions for interaction with BI systems

  - Applications tend to be industry-specific and have their own sets of data models
  - Thus shown as vertical entities
- Services Layer
  - Shown as horizontal framework across the vertical industry applications
  - At this layer, horizontal modules include
    - Physical network that IoT applications run on
    - Underlying management protocols
    - Hardware
  - Examples include backhaul communications via cellular, MPLS networks, VPNs, and so on

- Services Layer
  - Riding on top is the common services layer.
  - This conceptual layer adds APIs & middleware supporting third-party services & applications
- Network Layer
  - Communication domain for the IoT devices and endpoints
  - Includes devices themselves and the communications network that links them
  - Examples include wireless mesh technologies, such as IEEE 802.15.4, and wireless point-tomultipoint systems, such as IEEE 801.11ah

IoT World Forum (IoTWF) Standardized Architecture



- Defines set of levels with control flowing from the center (cloud service or dedicated data center), to the edge
- May includes sensors, devices, machines, and other types of intelligent end nodes
- In general, data travels up stack, originating from edge, and goes northbound to the center
- Using this model, we are able to achieve:
  - Decompose the IoT problem into smaller parts
  - Identify different technologies at each layer and how they relate to one another

- Define a system in which different parts can be provided by different vendors
- Have a process of defining interfaces that leads to interoperability
- Define a tiered security model that is enforced at the transition points between levels

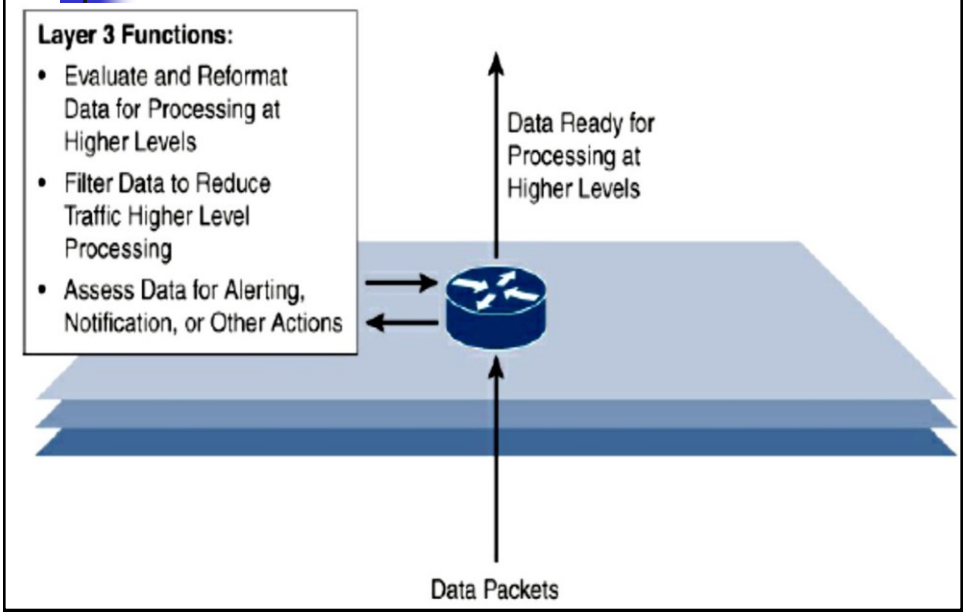
Layer 1: Physical Devices and Controllers Layer

- This layer is home to the "things" in the IoT, including various endpoint devices & sensors
- Size of these "things" can range from almost tiny sensors to huge machines in factory
- Their primary function is generating data and being capable of being controlled over network
- Sensor gathers the records however that we need to convert it into understandable format & join the one's sensor device using some protocol that we want to configure right here in layer two

Layer 2: Connectivity Layer

- Communications Between layer 1 devices
- Reliable delivery of information across the network
- Switching and routing
- Translation between protocols
- Network level security
- Network connectivity, join your tool with wi-fi connectivity or net stressed connection. This connectivity is changed based on context & area.

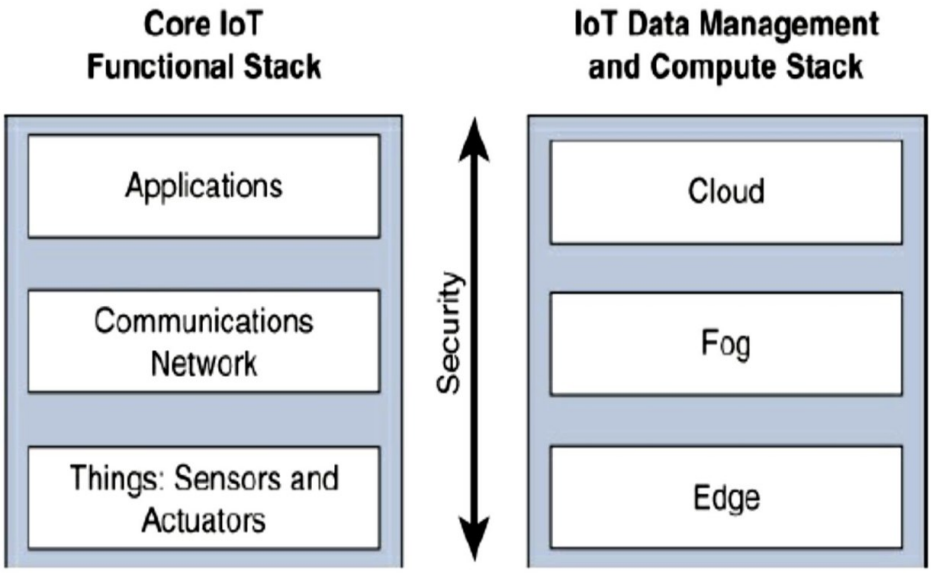
Layer 3: Edge Computing Layer



Upper Layers: Layers 4–7

IoT Reference Model Layer	Functions
Layer 4: Data accumulation layer	Captures data and stores it so it is usable by applications when necessary. Converts event-based data to query-based processing.
Layer 5: Data abstraction layer	Reconciles multiple data formats and ensures consistent semantics from various sources. Confirms that the data set is complete and consolidates data into one place or multiple data stores using virtualization.
Layer 6: Applications layer	Interprets data using software applications. Applications may monitor, control, and provide reports based on the analysis of the data.
Layer 7: Collaboration and processes layer	Consumes and shares the application information. Collaborating on and communicating IoT information often requires multiple steps, and it is what makes IoT useful. This layer can change business processes and delivers the benefits of IoT.

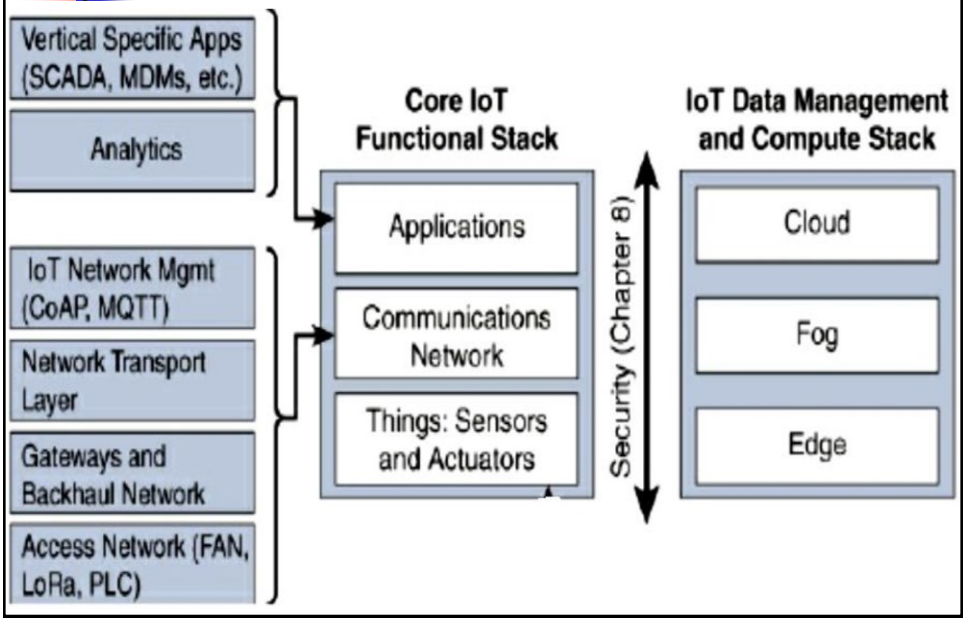
A Simplified IoT Architecture



A Simplified IoT Architecture

- Framework is presented as 2 parallel stacks:
  - IoT Data Management and Compute Stack
  - Core IoT Functional Stack
- Intention is to simplify the IoT architecture into its most basic building blocks

A Simplified IoT Architecture



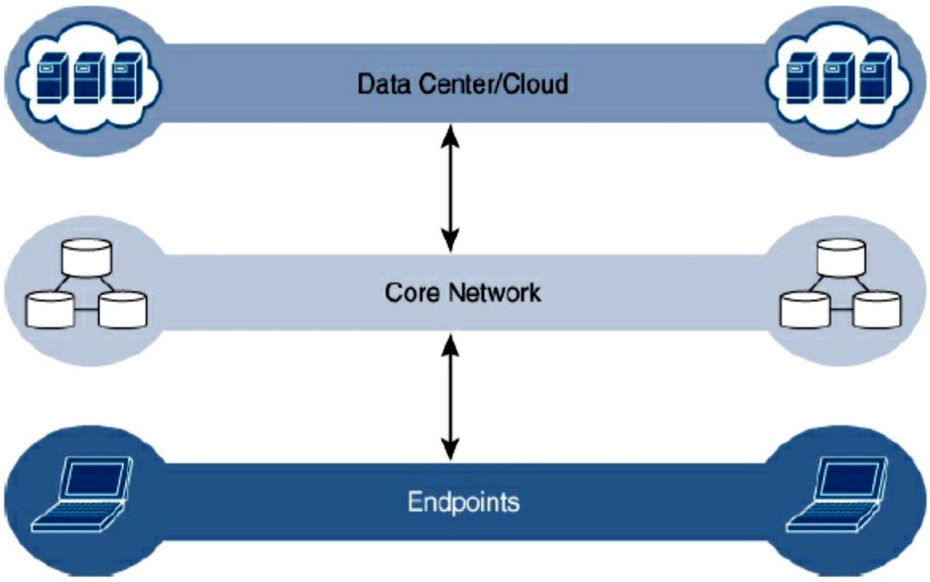
The Core IoT Functional Stack

- Every published IoT model include core layers, including "things," a communications network, and applications
- Framework presented here separates core IoT & data management in parallel & aligned stack
- Allows you to carefully examine functions of both, network & applications, at each stage of complex IoT system
- This separation gives you better visibility into the functions of each layer
- Presentation in 3 layers is meant to simplify your understanding of IoT architecture into its most foundational building blocks
- Such simple architecture needs to expanded

IoT Data Management and Compute Stack

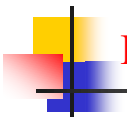
- The “things” connected to the Internet are continuing to grow exponentially
- Cisco predicted that by 2020 there will be more than 50 billion devices connected to some form of an IP network
- If number of devices is beyond conventional numbers, surely the data generated by these devices must also be of serious concern
-

IoT Data Management and Compute Stack



IoT Data Management and Compute Stack

- Data-related problems need to be addressed:
  - Bandwidth in last-mile IoT networks is very limited
  - Latency can be very high (Instead of dealing with latency in the milliseconds range, large IoT networks latency of hundreds to thousands of milliseconds)
  - Volume of data transmitted can be high
  - Big data is getting bigger



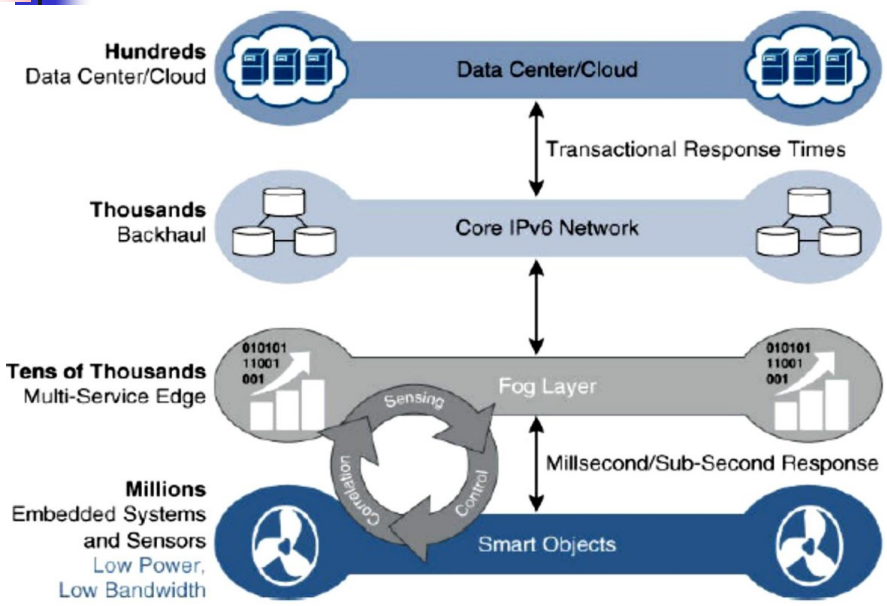
Fog Computing

- Solution to the various challenges is to distribute data management throughout the IoT system, as close to the edge of the IP network as possible
- Best-known example of edge services in IoT is fog computing
- Any device with computing, storage, and network connectivity can be a fog node
- Concept of fog was first developed by Flavio Bonomi and Rodolfo Milito of Cisco Systems
- In world of IoT, fog gets name from a relative comparison to computing in cloud layer
- Like clouds exist in sky, fog rests near ground
- In the same way, the intention of fog computing is to place resources as close to the ground—that is, the IoT devices—as possible

Fog Computing

- Examples : industrial controllers, switches, routers, embedded servers, and IoT gateways
- An advantage of this structure is that fog node allows intelligence gathering (analytics) and control from the closest possible point
- In one sense, this introduces new layer to the traditional IT computing model, one that is often referred to as the “fog layer”
-

Fog Computing



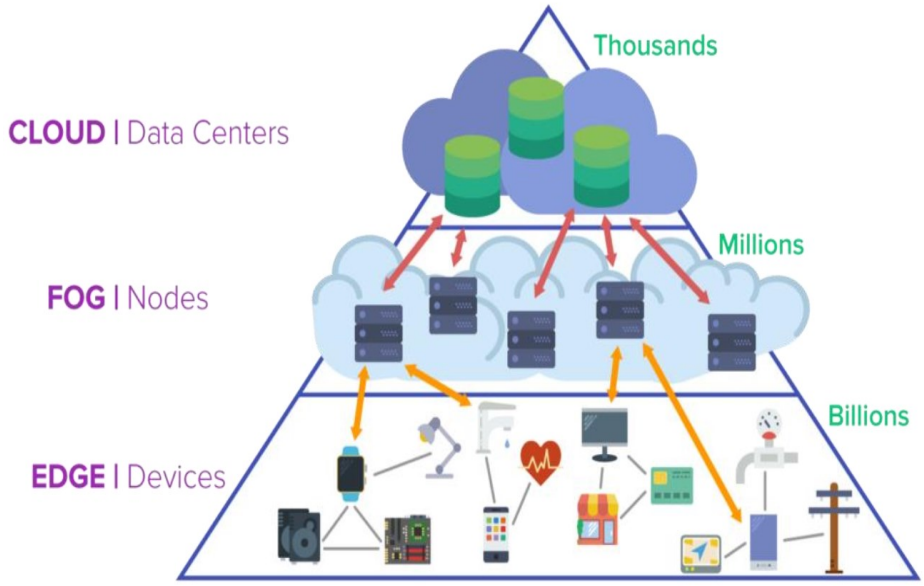
Edge Computing

- Also called as “mist” computing
- If clouds exist in sky, and fog sits near ground, then mist is what actually sits on the ground
- Thus, concept of mist is to extend fog right into IoT endpoint device itself
- Fog computing solutions are being adopted by many industries

Cloud vs Fog vs Edge

- Cloud computing dominated the networked systems landscape until recently
  - End-devices merely information gatherers
  - All intelligence in the cloud (relational DBs, analytics, web interfaces, control functions)
- As the number of devices grow, applications diversify and generate more data, this will not scale
  - Routing and storage costs
  - Signalling overheads
  - Latency inappropriate for real-time apps

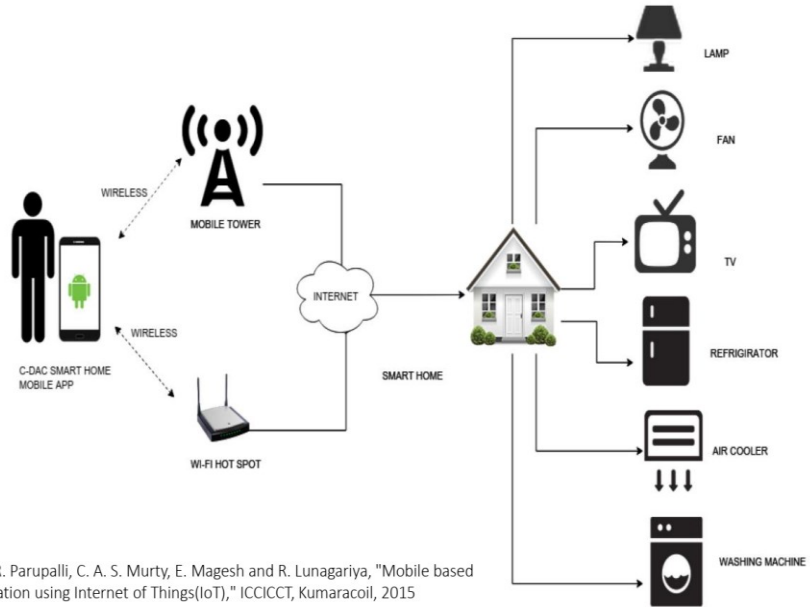
Ultimately it is about performance



Fog computing

- Pushing some of the effort closer to the device, i.e. to access networks/gateways.
- This includes data aggregation, compression, (partial) processing; making localised decisions.
- IoT device
  - Not required to be extremely smart, i.e. unique address and ability to communication directly with the cloud
  - No substantial storage
  - Battery powered (lifetime)

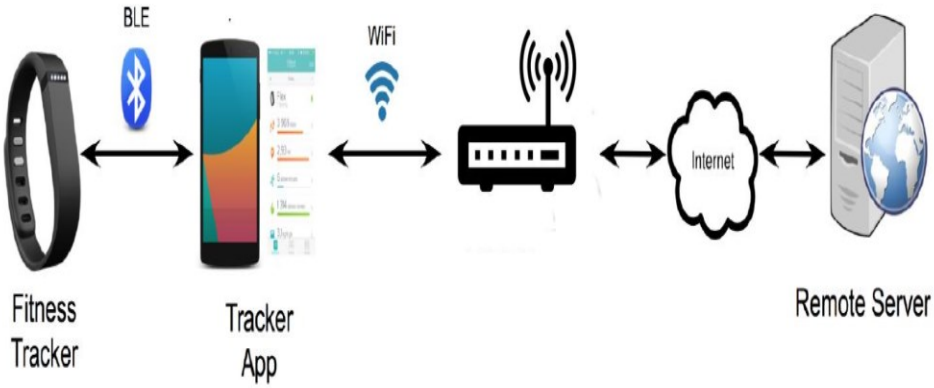
Example: Home automation

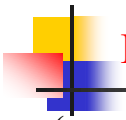


K. Mandula, R. Parupalli, C. A. S. Murty, E. Magesh and R. Lunagariya, "Mobile based home automation using Internet of Things(IoT)," ICCICCT, Kumaracoil, 2015

Example: Fitness tracking systems

- The user's mobile phone acting as gateway





Edge computing

- (process as much as possible where data is collected)
- Pushing processing power, communication capabilities, intelligence down at device level
- Emerging applications range from autonomous vehicles, to VR glasses, to earbuds.
- Do as much processing as required on the device, transmit only what is relevant long term or summaries
  - Low latency and decentralised decisions
  - Less signalling and communication overhead
  - Personalised experience

Example: Deep Learning at the Edge

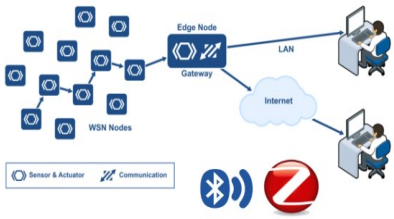
- Hardware Support: Low-power chips specialised in computationally intensive tasks (IBM TrueNorth,Movidius, Huawei Kirin)
- Software: lightweight inference frameworks optimised for constrained devices (mobile TensorFlow, Apple CoreML, DeepSense)
- Dedicated NN architectures: Model compression (SqueezeNet, MobileNet), point-wise group convolution (ShuffleNet), model pruning (NestDNN).

ABCD's of IoT

Applications

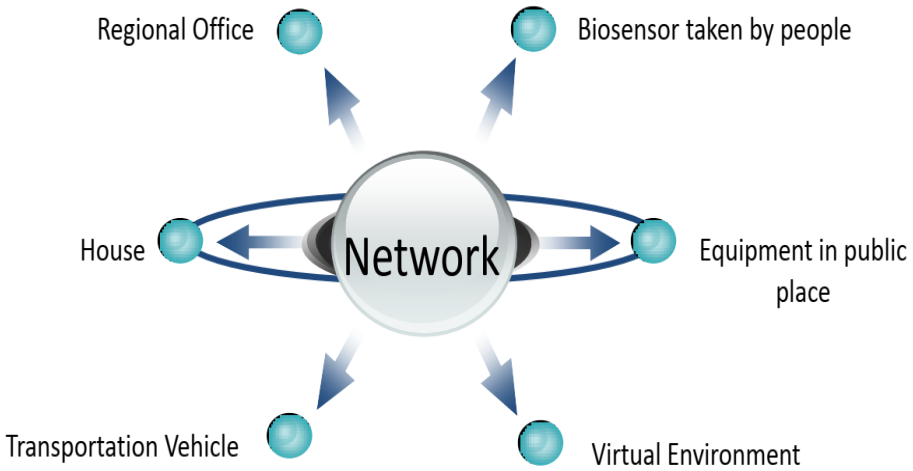
Big Data Analytics

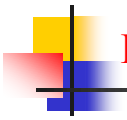
Connectivity and Communication



Devices – that are smart!

Applications





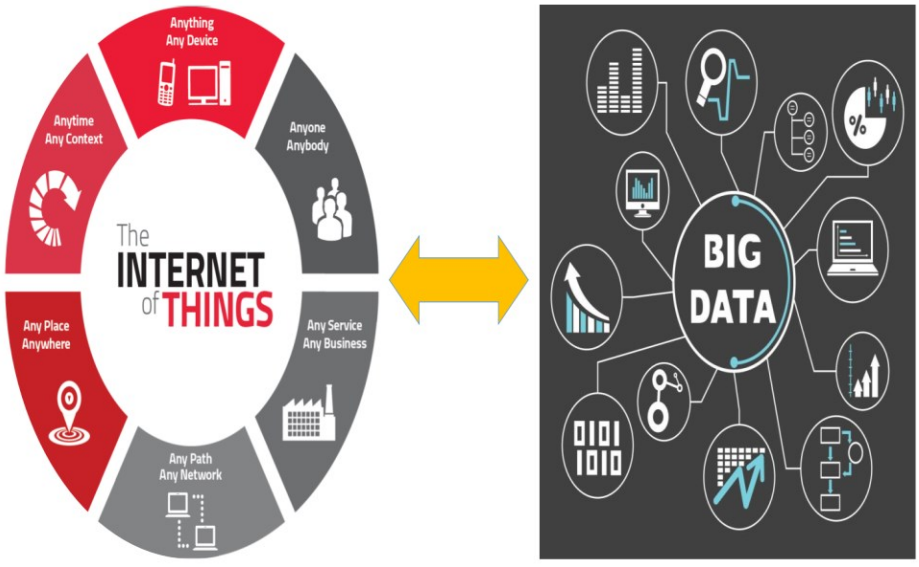
Big Data Analytics

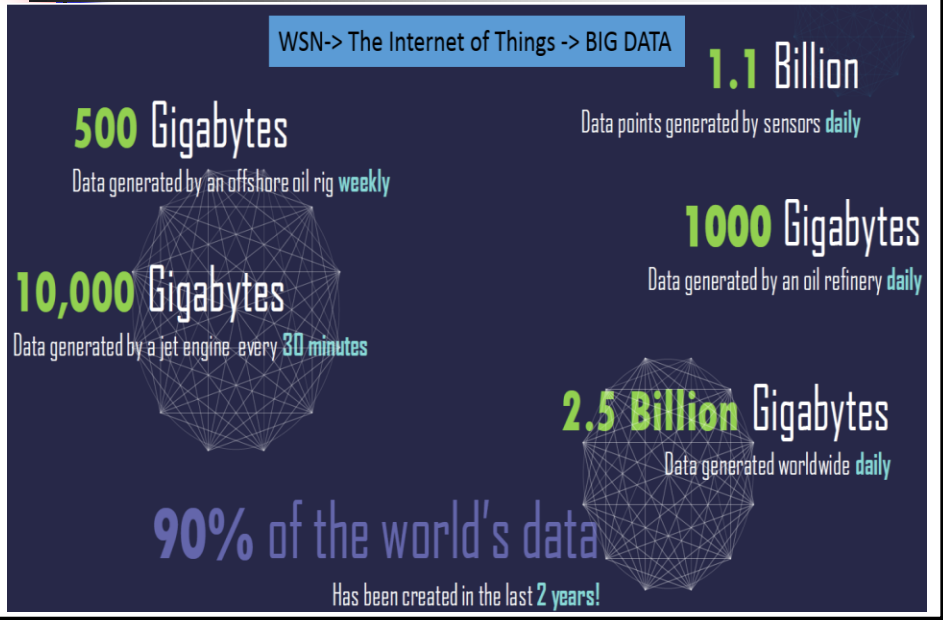
- Map-Reduce
- Frequent Item-sets
- Similarity
- Clustering
- Dimension Reduction
- Streaming Data

Why is there big data?

- Why is there big data?
- Number of devices increasing exponenEally
  - They conEnuously generate data
  - For example, on average, 72 hours of videos are uploaded to YouTube in every minute.

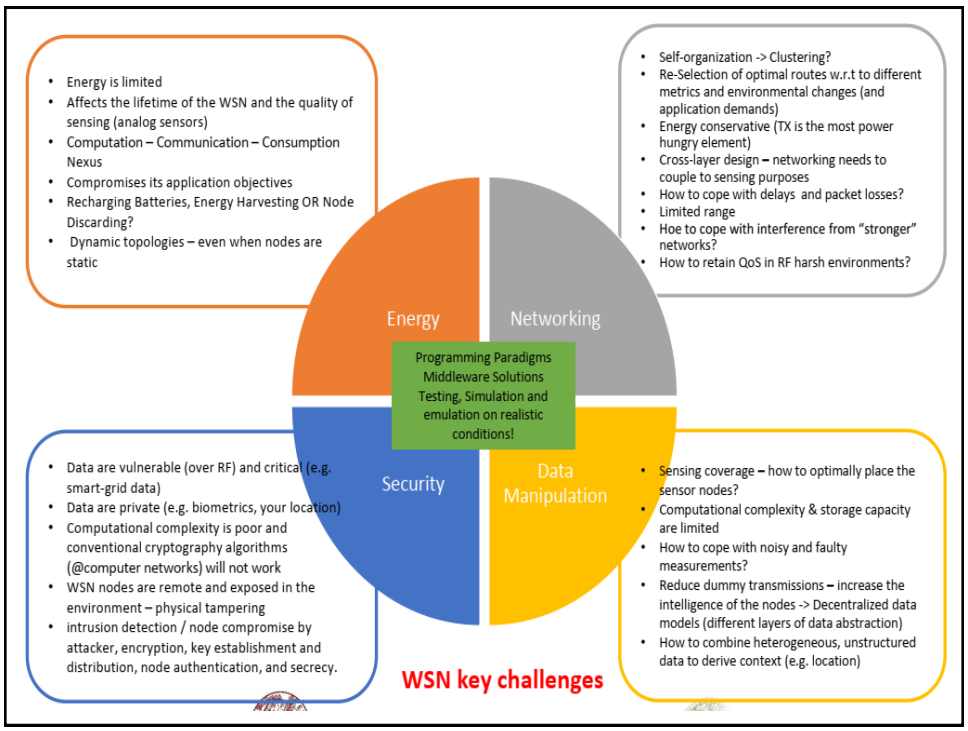
IoT Meets Big Data





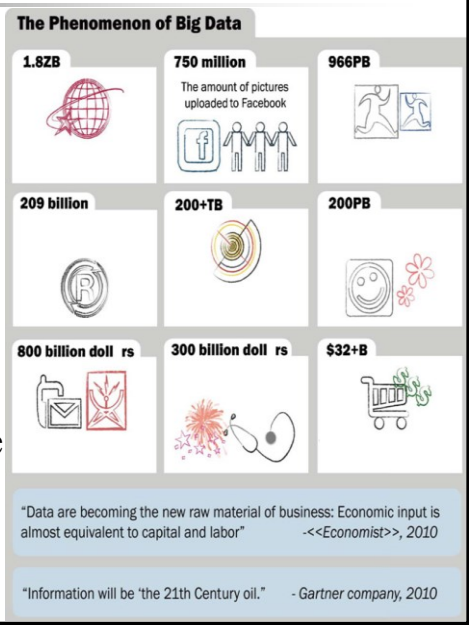
Big Data in WSN/IoT

- Big Data
  - **Volume**: size of data such as terabytes (TB), petabytes (PB), zettabytes (ZB),
  - **Variety**: types of data from difference sources (sensors, devices, social networks, the web, mobile phones)
  - **Velocity**: how frequently the data is generated (every millisecond, second, minute, hour, day,week, month, year.) Processing frequency may also differ from the user requirements.
- Challenges
  - High volume processing using low power processing architectures.
  - Discovery of real-time data-adaptive Machine learning techniques.
  - Design scalable data storages that provide efficient data mining



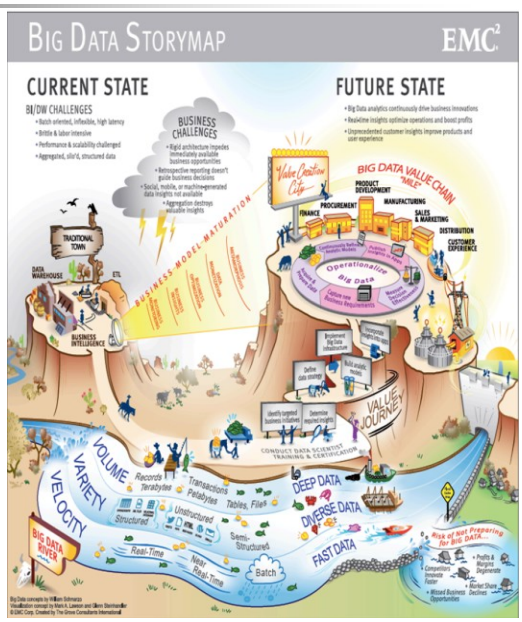
How much data is big?

- 2010: Apache Hadoop: "datasets which **could not be** captured, managed, and processed by **general computers** within an **acceptable scope**."
- **3V** model: Volume, Velocity, Variety [META] -+1V: Value [IDC]



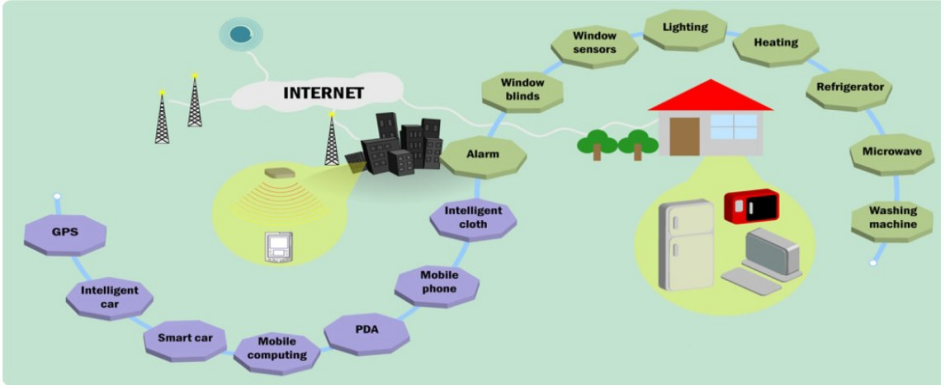
Value of Big Data

- New business and efficiency opportuniEs
- $300B in US medical industry
- Increased efficiency of government operaEons
- Search engines personalized for users
- Personalized ads, products, etc.



IoT and Big Data

- IoT applicaEons conEnuously generate data
- Even the smallest device generates data
- The problem: data processing capacity is lower than data generation speed

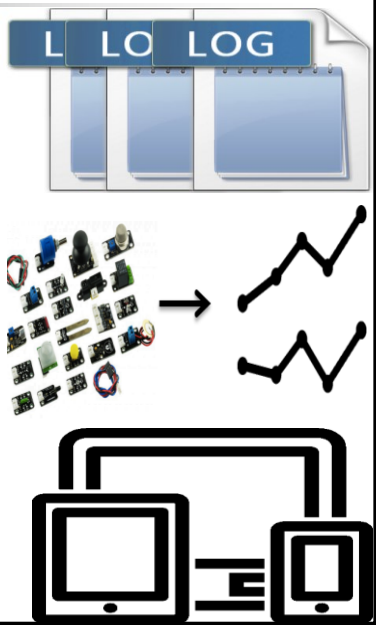


Data Generation

- Enterprise data: big companies, e.g. Facebook, Amazon
  - Business data is expected to double every 1.2 years
  - Walmart processes 1M customer trades/hour
  - Akamai adverEses 75M events/day
- IoT data: pervasive applicaEons, clinical medical care--R&D
  - Large scale, heterogeneous and strongly correlated data
  - 30 billion RFID tags and 4.6 billion camera phones are used around the world today
  - If Wal--Mart operates RFID on item level, it is expected to generate 7 terabytes (TB) of data every day
- Bio--medical data: human gene sequencing
  - One sequencing of human gene may generate 100 sequences of 600GB raw data
- Other areas: physics, bio--informaEcs, etc.
  - Astronomy: Sloan Digital Sky Survey (SDSS), the data volume generatd per night surpasses 20TB 8

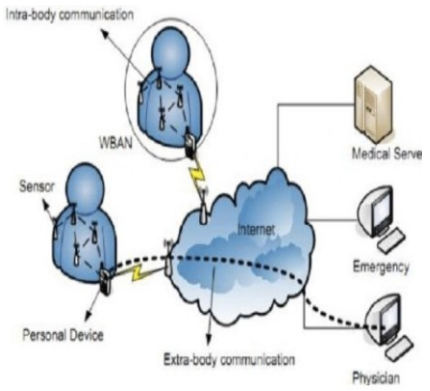
Data Acquisition

- Log files: almost all digital devices provide logging capability
- Web activity recording, financial applications, network monitoring
- Sensing: physical quantities into readable digital signals
- Sound wave, voice, vibration, automobile, chemical, current, weather, pressure, temperature, etc.
- Localization
- Mobile plakorms: similar to sensing
- More personalized, specific to a user



Data Transportation

- Data transfer to a storage infrastructure for processing and analysis
- Inter data center network (DCN) transmissions:
  - Source to data center
  - Using WAN: 40--100Gbps
- Intra DCN transmissions:
  - Data center interconnect
  - Top--of--the--rack vs.aggregator switches
  - 1--10--100 Gbps

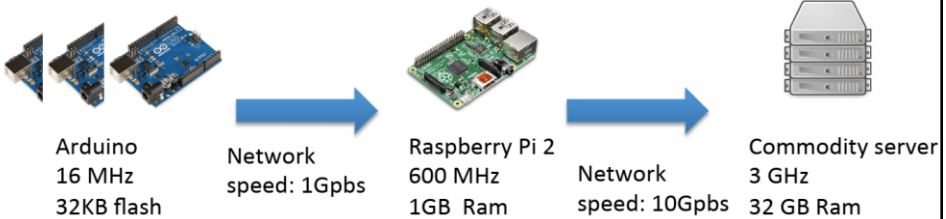


Data Preprocessing

- Eliminate or reduce redundancy, noise, meaningless data
  - Increase storage efficiency, data analysis speed
- IntegraEon: combining data from different sources
  - Data warehouse: ETL (Extract, Transform and Load)
  - Data federaEon
  - Mostly used by search engines
- Cleaning: how can data be cleaned?
  - Define error types --> idenEfy errors --> correct errors --> document errors --> modify infrastructure to prevent errors
- Redundancy eliminaEon
  - Redundancy detecEon, data filtering, data compression
  - Areas: Images, videos
- One solution: Compression!

Preprocessing Capabilities

- Assume there is a job with 1TB total size
- 100K Arduino, 1K Raspberry Pi 2, 100 servers
- Time spent in computaEon vs. networking
  - Ardunio level
  - Raspberry Pi 2 level
  - Server level

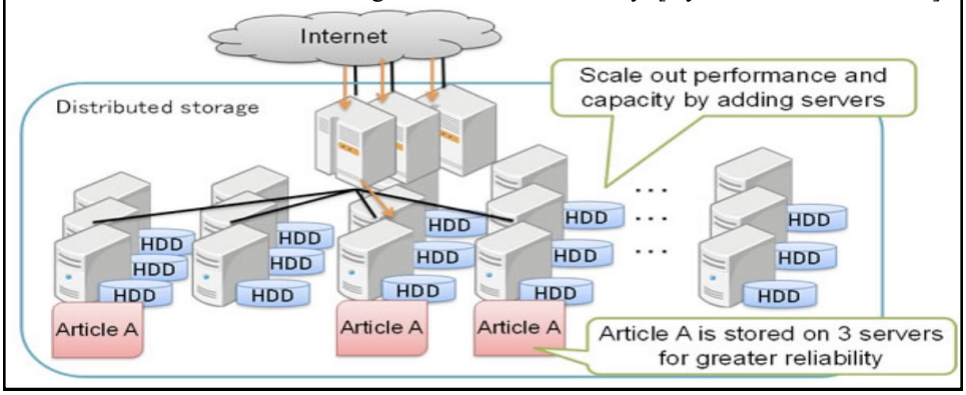


Big Data Storage

- Storage and management of **large--scale** data sets while achieving **reliability** and **availability** of data accessing
- TradiEonally on servers with structured RDBMSs.
- ExisEng storage systems for massive data
- Direct asached storage (DAS)
- Several hard disks directly connected with servers
- Only suitable to interconnect servers with a small scale
  - Network asached storage (NAS)
- NAS uElizes network to provide a union interface for data access and sharing
- The I/O burden is reduced extensively since the server accesses a storage device indirectly through a network
  - Storage area network (SAN)
- Designed for data storage with a scalable and bandwidth intensive network
- Data storage management is relaEvely independent within a storage local area network 13

Distributed Storage System

- CAP: Consistency Availability Partition tolerance
  - At most two of the three requirements can be satisfied simultaneously
- CA vs. CP vs. AP systems
  - CA: for single servers
  - CP: useful for moderate load [BigTable and Hbase]
  - AP: useful when no high demand on accuracy [Dynamo and Cassandra]



Connectivity

- M2M
- Wireless Sensor Networks
- IPv6 and 6LowPAN
- Bluetooth LE and ZigBee
- WiFi and LTE
- Backscatter

Devices and Platforms

- Mobile Systems
- Sensor Systems
- Wearables
- Energy Harvesting
- Security and Privacy