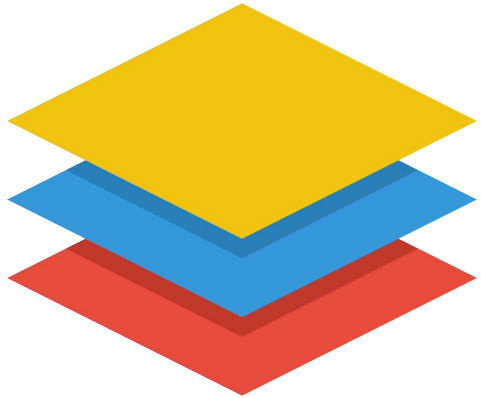
BAIT 2123
INTERNET OF THINGS

Chapter 2: Common Structure of IoT System

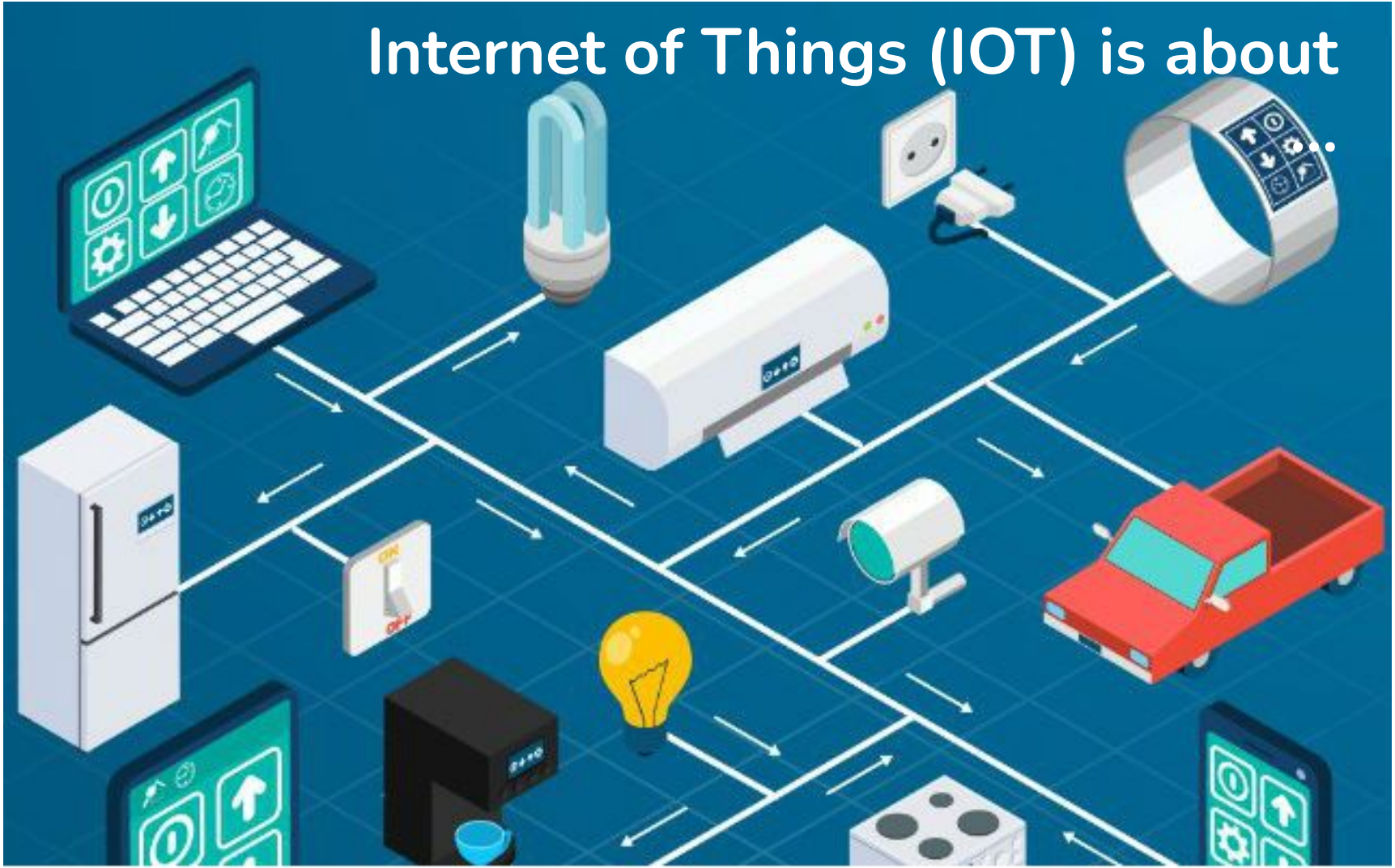
IoT Development and IoT Stack:
Application, Data Processing (Softtware Backend),
Network (Communication), Sensing (Hardware)

Overview

- IoT Development
  - Technology Infrastructure
  - Platforms Architecture
- IoT Stacks
  - Device Layer
  - Communication Layer
  - Core Platform Layer
  - Analytics Platform Layer
  - Solutions Layer



Internet of Things (IOT) is about ...



IoT Technology Infrastructure

A bird-eye view of the FOUR (4) major building blocks of IoT:-

- **Applications**
- **Software backend**
- **Communication**
- **Hardware**
- **Security** (worthwhile mentioning)

The IoT technology architecture is **currently far from being standardized** or accurately defined and it is evolving very quickly.

There are hundreds of different hardware units, connection protocols, low-level software languages, and an increasing number of IoT platforms. It is a relatively young infrastructure.

IoT Technology Infrastructure (con't)

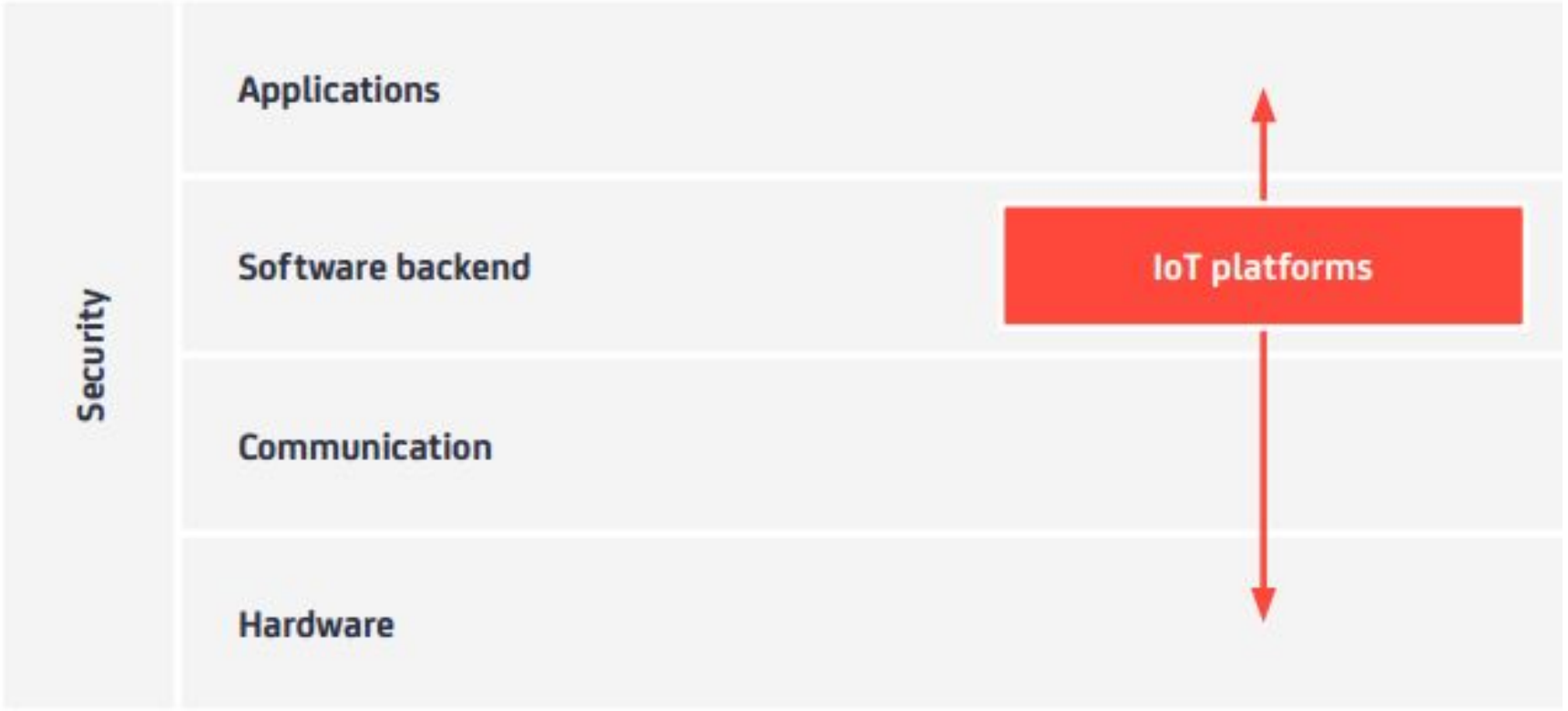


EXHIBIT 3: **Central building blocks of IoT** – IoT platforms are part of the central software backend in the IoT infrastructure (Source: IoT Analytics)

IoT Platform

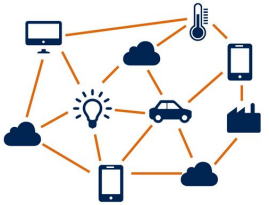
- An IoT Platform facilitates communication, data flow, device management, and the functionality of applications.
- IoT Platform links machines, devices, applications, and people to data and control centers.
- It employs better, quicker search engines and data storage systems with the capacity and sophistication to handle volume far beyond what has brought industry to the present moment.
- Most of its elements are **cloud-based** and running on **wireless connectivity**, which may be established via third-party providers, application programming interfaces (APIs), cellular capabilities, or a combination of these technologies.

Source: Matthew J. Perry. (2016). Evaluating and choosing an IoT Platform. O'Reilly.

IoT Platform (con’t)

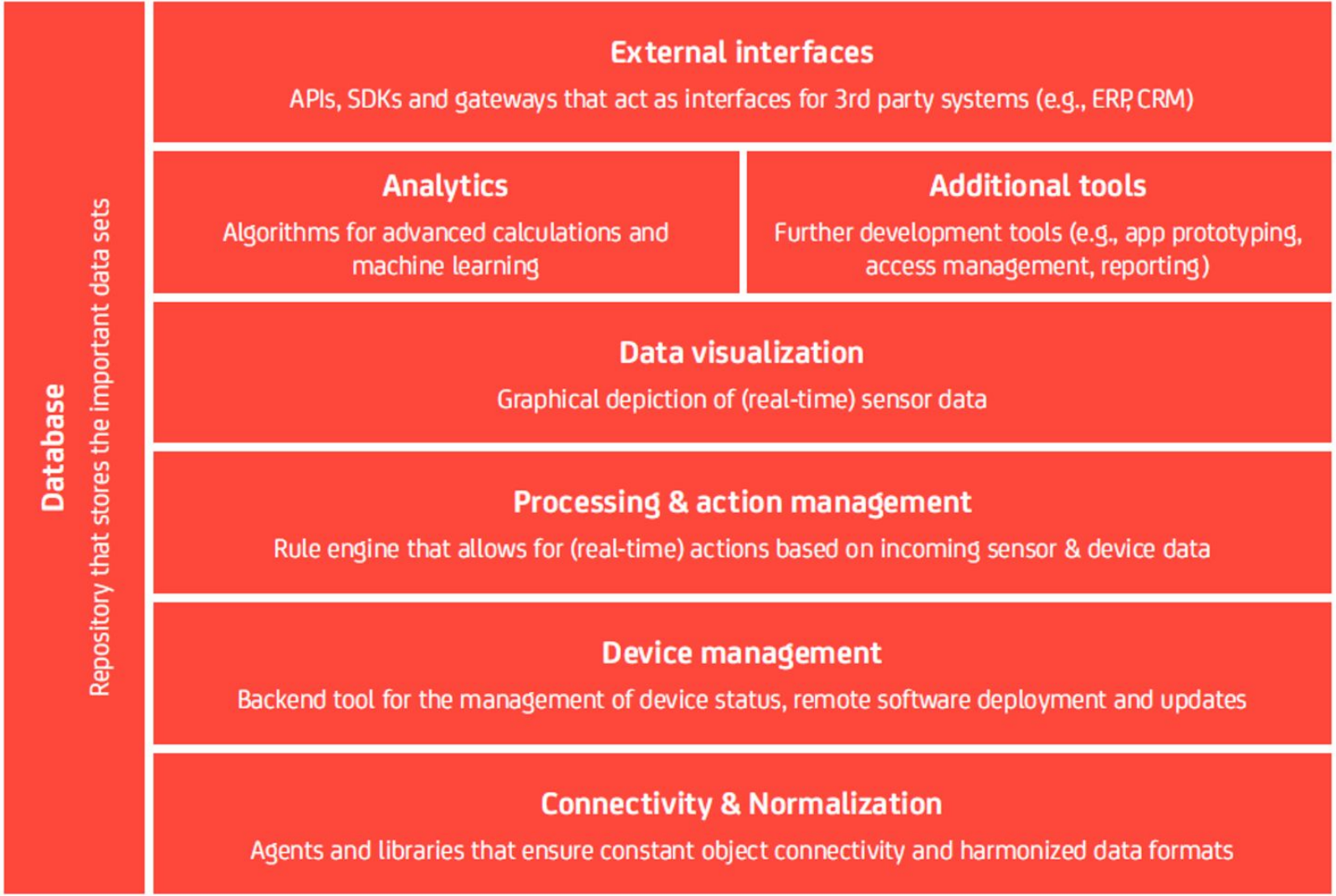
- A platform enables all these elements to **connect**, **monitor**, and **communicate** with each other with far greater speed and flexibility.
- Data from an ever-expanding **ecosystem** can be collected, sorted, and harnessed entirely online.
- Platform also enables **data prioritization**, ample **security** features, **scalability**, and abundant capacity for pulling in, **storing**, and **analyzing data**.

A modern IoT Platform architecture



- In its simplest form, an IoT platform is just about enabling connectivity between "*things*" or devices. The architecture may also consist of a software platform, an application development platform or an analytics platform. In a more sophisticated form, a **true end-to-end IoT platform** consists of eight important architectural components:
  - **Connectivity & normalization**
  - **Database**
  - **Analytics**
  - **Additional Tools**
  - **Device management**
  - **Processing & action management**
  - **Visualization**
  - **External interfaces**

Source: Padraig Scully. (2016). 5 things to know about the IoT ecosystem. IoT Analytics, Making insights for the Internet of Things.
Available at: https://iot-analytics.com/5-things-know-about-iot-platform/
Accessed on: 6th May 2017


Database
Repository that stores the important data sets
External interfaces
APIs, SDKs and gateways that act as interfaces for 3rd party systems (e.g., ERP, CRM)
Analytics
Algorithms for advanced calculations and machine learning
Additional tools
Further development tools (e.g., app prototyping, access management, reporting)
Data visualization
Graphical depiction of (real-time) sensor data
Processing & action management
Rule engine that allows for (real-time) actions based on incoming sensor & device data
Device management
Backend tool for the management of device status, remote software deployment and updates
Connectivity & Normalization
Agents and libraries that ensure constant object connectivity and harmonized data formats

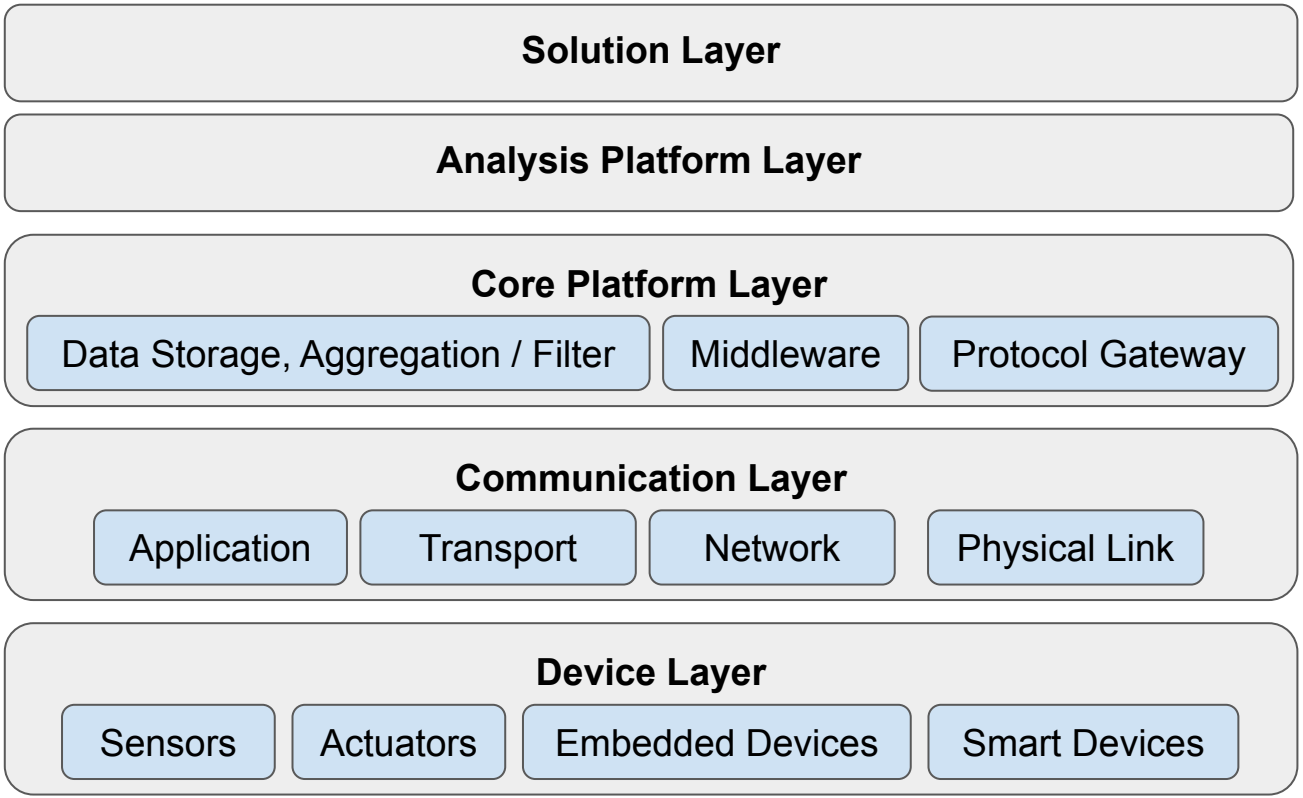
A modern IoT Platform architecture

- **Connectivity & normalization:** brings different protocols and different data formats into one "*software*" interface ensuring accurate data streaming and interaction with all devices.
- **Device management:** ensures the connected "*things*" are working properly, seamlessly running patches and updates for software and applications running on the device or edge gateways.
- **Database:** scalable storage of device data brings the requirements for hybrid cloud-based databases to a new level in terms of data volume, variety, velocity and veracity.
- **Processing & action management:** brings data to life with rule-based event-action-triggers enabling execution of "*smart*" actions based on specific sensor data.

A modern IoT Platform architecture

- **Analytics:** performs a range of complex analysis from basic data clustering and deep machine learning to predictive analytics extracting the most value out of the IoT data-stream.
- **Visualization:** enables humans to see patterns and observe trends from visualization dashboards where data is vividly portrayed through line-, stacked-, or pie charts, 2D- or even 3D-models.
- **Additional tools:** allow IoT developers prototype, test and market the IoT use case creating platform ecosystem apps for visualizing, managing and controlling connected devices.
- **External interfaces:** integrate with 3rd-party systems and the rest of the wider IT-ecosystem via built-in application programming interfaces (API), software development kits (SDK), and gateways.

IoT Stack - Reference Architecture in IoT



Device Layer

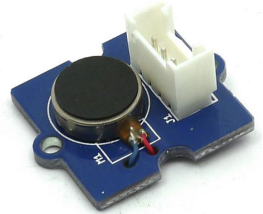
- Device are the “things” of Internet of Things.
- Every device around us has a potential to connect to the Internet and provides some useful information.
- The bottommost layer lays the end devices, edge devices and gateways.
- This layer also includes apps that can be installed in the end devices to enhance their functionality.
- Examples:
  - Sensors
  - Actuators / Indicator
  - Embedded Devices (Controller)
  - Smart Devices

Device Layer - Sensors

- Detect events and read changes in its environment.
- It can be programmed to sense the required environmental parameters that need to be monitored and convert them into meaningful data.
- Connected sensor
  - Capture information from the **physical resource** within a particular area and send it across over the network.
  - E.g., smart water sensors measure quality of water and pollution levels by sensing polluting ingredients in the water.
- Single form sensor
  - **Standalone**, emits signals based on the target object's movement
  - Makes as an alert beep at regular intervals.
  - E.g., Medicine bottle with sensors to alerts with a beep to patient for taking medicine on time.

Device Layer - Actuators

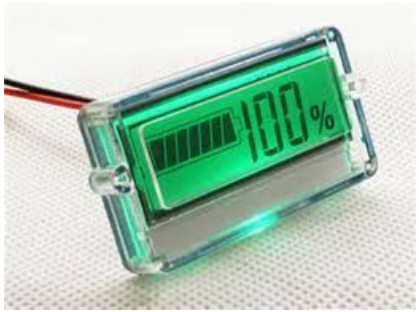
- Act with the physical world based on the input received and carry out the required action.
  - DC / AC servo motor, Step motor
  - E.g., Connected door locks controlled using remote control device. Actuator being responsible for controlling the movement of the lock motor.

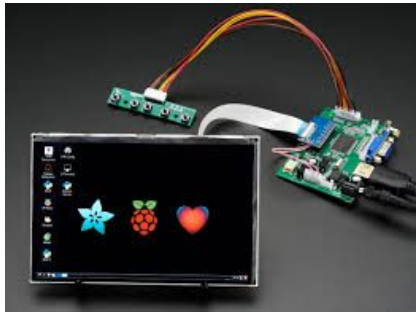


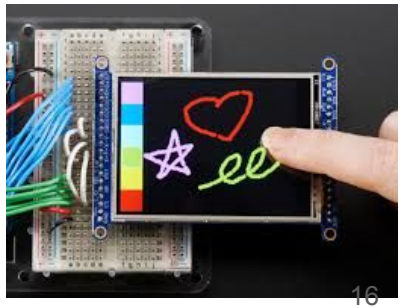
Device Layer - Actuators vs Indicators

Actuator = to execute

Indicator = to show

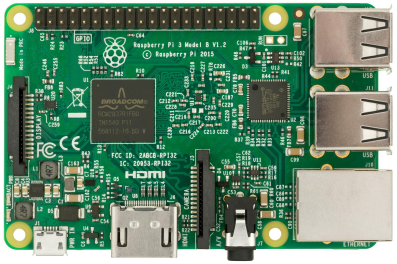


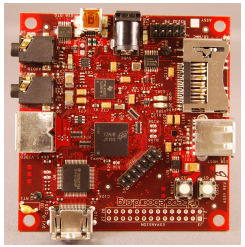




Device Layer - Embedded System (Controller)

- Prototype for testing IoT use cases.
- Well-known controller boards:
  - Arduino
  - Raspberry Pi
  - BeagleBone
  - …

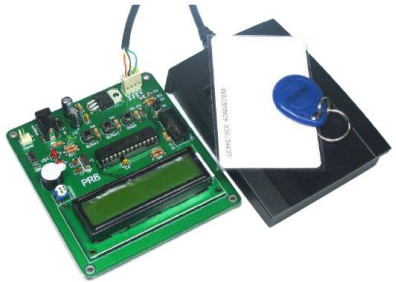




- Connects with different sensors and actuators and provides network ports for internet connectivity.
- Availability of controller becomes an ideal prototyping choice, and leads to build a commercial connected product.

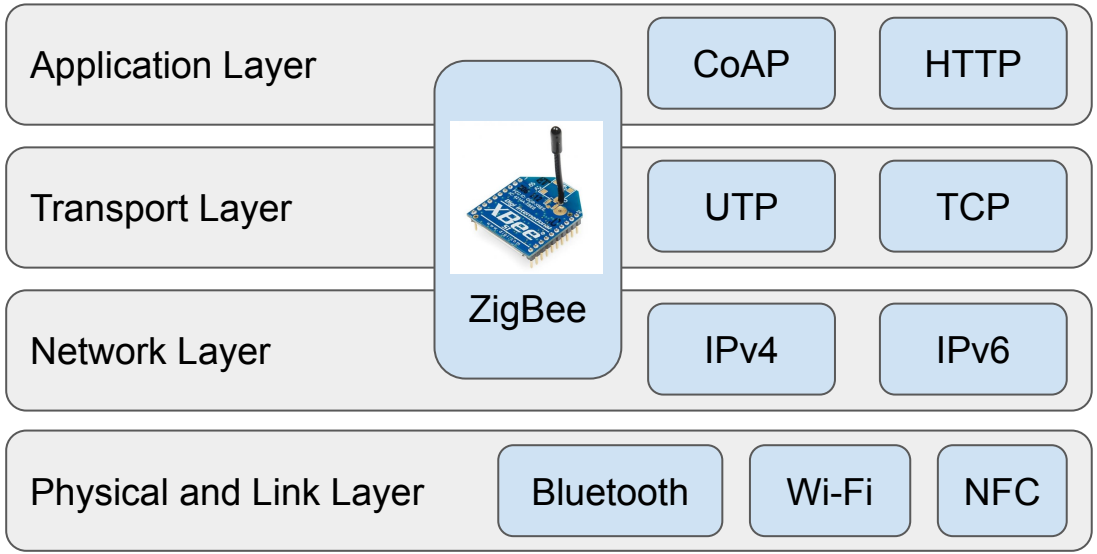
Device Layer - Smart Devices

- Price to build controller boards have fallen down dramatically, makes controller can be envisioned to embed in any object and more real-time taking object.
- Smart devices such as smart mobile phone, smart tablets, smart watch are included as instances of embedded devices.
- Healthcare Industry:
  - Glucometer, heart rate, blood pressure monitors, imaging systems…
- Automotive Industry:
  - Tire pressure monitor, connected transportation services...
- Supply Chain Industry:
  - RFID tagging for efficient tracking...
- Energy and Utilities Industry:
  - Smart Grids, Smart Meter...



Communication Layer

- This layer utilizes devices in device layer for communicating among themselves, and between them and the cloud.



Various Protocol in IoT Communication Layers

CoAP - Constrain Application Protocol

NFC - Near Field Communication

Communication Layer

- Communication layer is discussed based on type of devices with different network abilities (strategy and protocols).

- Communication Strategy
  - Gateway
    - Device
    - Smart (Edge)
    - Smartphone
  - Connectivity
    - Direct
    - Device-to-device
    - API Connectivity

- Communication Protocol
  - Wi-Fi
  - Ethernet
  - Cellular
  - Bluetooth Low Energy
  - RFID
  - NFC
  - ZigBee / Z-wave

- Application Protocol
  - MQTT
  - CoAP
  - AMQP
  - WebSocket
  - AllJoyn
  - DDS

- Industrial Protocol
  - BACnet
  - SCADA
  - Modbus

Communication Strategy - Gateway

- Device gateway connects incompatible networks or protocol and provides a means to connect devices to the Internet.
  - E.g., Sensors connect to a gateway through Bluetooth and communicate with the broadband router using Wi-Fi to connect to Internet.
- Smart gateway (edge gateway) has own local storage and embedded application to perform data analytics directly from the devices.
  - E.g., “Fog computing” - Collected sensor data need not be sent over the network to the Internet and can be analyzed or filter close to the device in the gateway itself.
- Smartphone as a gateway, with the capability of Bluetooth protocol, cellular network, sensors (accelerometers, gyroscope, magnetometer…).
  - Manual intervention is required to set up communication process.

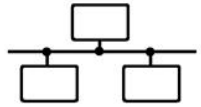
Communication Strategy - Connectivity

- Direct Connectivity to Internet via Wi-Fi, Cellular Network, especially for cloud services.
- Device-to-device connectivity to form a mesh network.
  - To collect information, report their existing states, alertness, perform discovery routines.
  - E.g., Home security system connects to nearby alarm system to alert if someone approaches a door.
- Application Program Interface (API) as a REST URL to communicate to another service provider, then triggers action to another device.
  - Eliminates the need of installing a shared gateway
  - For multiple disparate devices associated with a vendor service.

*REST - REpresentational State Transfer
*URL - Uniform Resource Locator

Communication Layer - Communication Protocol

- Devices connect to Wi-Fi gateway securely using security options.
  - Wi-Fi devices consumes a lot of power, becomes an overhead for small constrained devices.
- Ethernet is a wired option for network communication



- Cellular requires a data subscription package from network provider.
  - E.g., 2G, 3G, 4G, LTE, etc

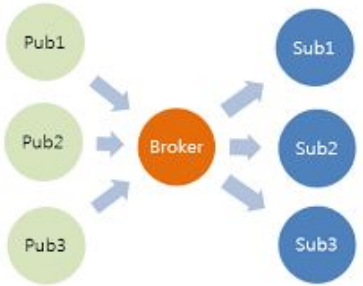
- Connection among Bluetooth devices instantly and send data to the cloud
  - With limited distance ~100 meters

- Radio Frequency IDentity communicates data between tags and readers.
  - E.g., Track and trace assets or products at entry and exit points.
- Near Field Communication (NFC) act as tag or reader
  - Establish peer to peer communication between devices, uses 13.56 MHz.

- In general, ZigBee uses 2.4GHz frequency to form a mesh network topology.
  - It requires a gateway device for Internet routing.

Communication Layer - Application Protocol

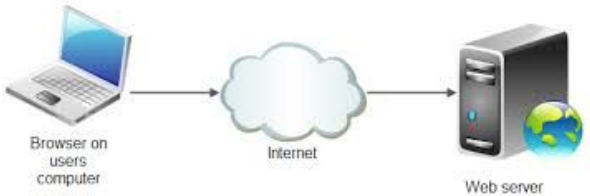
- Message Query Telemetry Transport (MQTT)
  - Lightweight message oriented middleware based on publish and subscribe model, with hierarchical topic(s) oriented.
  - Devices connects to brokers over TCP protocol to publish or subscribe based on a topic.
  - Advantages:
    - Support low bandwidth network.
    - Topic definition supports taxonomy specific design on particular domain.
- Constrained Application Protocol (CoAP)
  - Optimized for constrained power and processing capabilities of IoT.
  - UDP (message) and HTTP (request / response interactions)
  - Acknowledgement, Reset, Confirmable, Non-confirmable.
  - Advantages:
    - Cost effective on power consumption / high data volume communication
    - Cost-wise in data push and pull mode
    - Reference for RESTful API and CoAP



Communication Layer - Application Protocol

- Advanced Message Queuing Protocol (AMQP)
  - Message oriented middleware supports queue based and publish / subscribe model.
  - Operates on TCP transport and interoperable between client and server.
    * Reference: https://www.amqp.org/about/members

- WebSocket
  - Standardized by World Wide Web Consortium (W3C) and Internet Engineering Task Force (IETF).
  - Supports a persistent connection between client and server over a single TCP connection.
  - Facilitates near real-time communication without polling / explicit request for data updates.



Communication Layer - Application Protocol

- AllJoyn
  - A collaborative open-source software framework that allows devices to be discover, publish / broadcast and communicate with each other.
  - Promotes interoperability and seamless integration between devices and application via a set of core features.
    - E.g., Android to iOS, Linux, OpenWRT Windows, OS X.
    - Reference: https://github.com/alljoyn/alljoyn.github.com/wiki

- Data Connectivity Standard (Data Distribution Services)
  - A part of Object Management Group (OMG) IoT Standards.
  - Enable Interoperability between machines, enterprise system and mobile devices.
  - Reference: https://www.dds-foundation.org/what-is-dds-3/

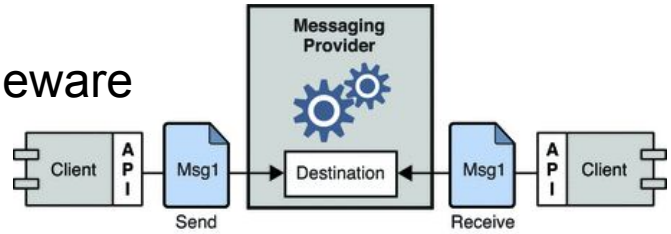
Communication Layer - Industry Protocol

- BACnet
  - Enables Smart Building automation by providing infrastructure for application to monitor and control network of BACnet devices.
  - Information exchange between devices to handle communication in objects (device information), service (access, management, event, transfer and virtual terminal) and transport (channel and protocol) area.
- Supervisory Control and Data Acquisition (SCADA)
  - Enables remote control and monitoring of industrial devices to provide automation.
  - Includes Remote Terminal Units (RTU) / Programmable Logic Controller (PLC) as specialized processing units to reads sensors data and process it into digital format, and sent it to Human Machine Interface (HMI) panels for the end user to monitor the industrial process flow.
- Modbus
  - A serial communication protocol using standard RS-485 and RS-232, used in SCADA network.
  - Modbus RTU is conceptualized on master/slave model
  - Modbus TCP is a modern approach with greater throughout compared to RTU.

Core Platform Layer

- This layer contains the cloud services and resources that **support** resource management (in **Analytic Platform Layer**) and processing of IoT task that reach the cloud.
- There are **no standardized protocols** where all vendors can converge on. Therefore, IoT stack needs to provide support for commonly used protocols, industry protocol and any evolving standards in future.
- Protocol Gateway is required to **convert** proprietary **protocol** to the protocol supported by the platform **for communication**.
    - In IoT stack, protocol gateway provides connectivity to the devices over the messaging platform or middleware like MQTT or AMQP.

Core Platform Layer - Messaging Middleware



- It has been used as a **backbone for message communication** between enterprise system, as an integration pattern with various distributed system.
- Provides a highly scalable and performance middleware to **accommodate vast number** of ever growing **connected devices**
  - 4.9 billion connected things (2015) - expected to reach 25 billion in 2020.
- **Device management** includes devices registration, storage of device data, secure connectivity and accessibility by authorized personnel.
- Holds device data for the specific interval and in-turn a dedicated storage service is used to **scale, compute and analyze information**.

Core Platform Layer - Data Storage

- Deals with **storage of continuous data stream** from devices
- Requirements: Highly scalable storage service (**Terabytes**), enable faster data retrieval.
- **NoSQL databases** (e.g., MongoDB, Cassandra) enables faster retrieval, perform computations and eases processing for data analysis.
  - E.g., Connected car with ID field and speed value. ID is fixed but speed value is changed in every second. Values are collected over a period of time using Time Series database.
- For enterprise IoT application, **structured, semi structured and unstructured information** is recorded to correlate and derive insights.
  - E.g., Equipment manual information (unstructured) can be fed into the system and sensor data (structured) can be correlated with the manuals for raising functional alerts and suggest corrective measurements.

Core Platform Layer - Data Aggregation and Filter

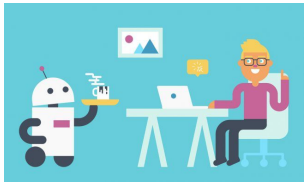
- When deals with raw data, **not all data** needs to be consumed and **treated equally** by your application.
- Device gateway may not have high computational power to filter out all scenarios. Therefore, in IoT application requires a design of **data filter component** based on your data dependency graph to filter incoming data.
- **Contextualize data** with more information (e.g., aggregating device data with asset management software to retrieve warranty information of physical devices)

Analytic Platform Layer (Software-Defined Resource Management)

- Flow and Task placement
  - This component keeps track of the state of available cloud, fog and network resources (information provided by the Monitoring Service) to identify the best candidates to hold incoming tasks and flows for execution.
  - This component communicates with the Resource-Provisioning service to indicate the current number of flows and tasks, which may trigger new rounds of allocations if deemed too high.
- Knowledge Base
  - This component stores historical information about application demands and resource demands that can be leveraged by other services to support their decision-making process.
- Raw Data Management
  - This service has direct access to the data sources and provides views from the data for other services. Sometimes, these views can be obtained by simple querying (e.g., SQL or NoSQL REST APIs), whereas other times more complex processing may be required (e.g., MapReduce). Nevertheless, the particular method for generation of the view is abstracted away from other services.

Analytic Platform Layer (Software-Defined Resource Management)

- Performance Prediction (Machine Learning)
  - This service utilized information of the Knowledge-based service to estimate the performance of available cloud resources. This information is used by the Resource-Provisioning service to device the amount of resources to be provisioned, in times where there are a large number of tasks and flows in use or when performance is not satisfactory



- Monitoring & Profiling (Actionable Insights)
  - This service keeps track of the performance and status of applications and services, and supplies this information to other services as required.
  - This service also builds resource- and application-profiles based on information obtained from the Knowledge-Based and Monitoring Services.

Analytic Platform Layer (Software-Defined Resource Management)

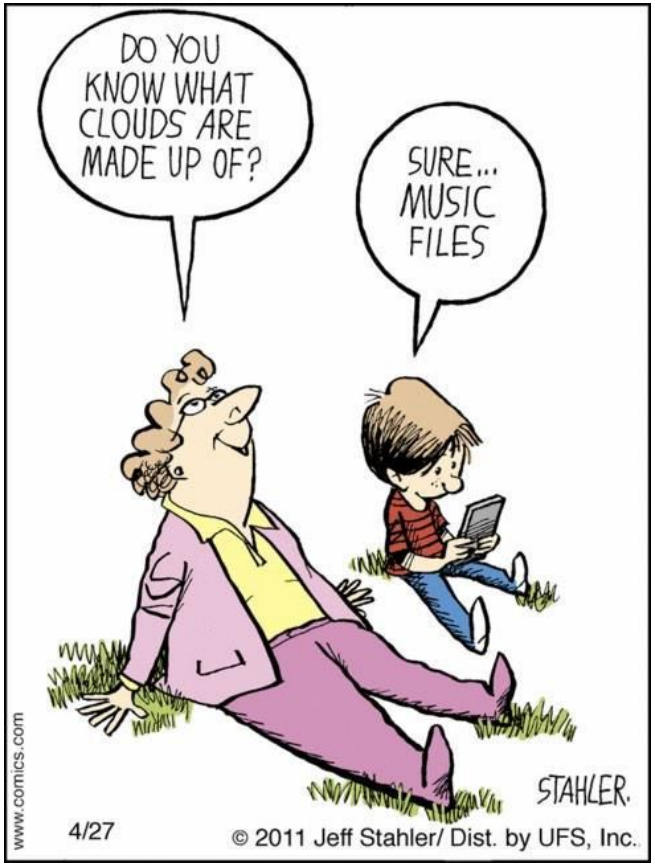
- Resource Provisioning
  - This service is responsible for acquiring cloud, fog and network resources for hosting the applications. This allocation is dynamic, as the requirements of applications, as well as the number of hosted applications, changes over time. The decision on the number of resources is made with the use of information provided by other services (such as Profiling, Performance Prediction and Monitoring), and user requirements on latency, as well as credentials managed by the Security service.
  - E.g., the component pushes tasks with low-latency requirements to edge of network as soon as free resources are available.
- Security
  - This service supplies authentication, authorization and cryptography, as required by service and application.

Solution Layer

- Top most layer contains the applications that leverage fog computing to deliver innovative and intelligent application to end users.
- Divided into two parts:
  - Solution Template: Includes Device Model, SDKs, Application Model, Composite IoT Services, Machine Learning Models and Event Services.
  - IoT Application: Customized application that requires common set of tasks and services, configurations of components via dashboard or APIs and the entire process can be simplified and abstracted through the use of solution templates.
- Example:
  - In connected car solution, abstract data model is made by Vehicle's runtime data, GPS data and asset data of the vehicle. This data model provides solution templates irrespective of model for remote monitoring, predictive maintenance or geospatial analysis.

Summary

- Device Layer
- Communication Layer
- Core Platform Layer
- Analytics Platform Layer
- Solutions Layer


DO YOU KNOW WHAT CLOUDS ARE MADE UP OF?
SURE... MUSIC FILES
STAHLER.
www.comics.com
4/27
© 2011 Jeff Stahler/ Dist. by UFS, Inc.