L2: System Overview of Smart Objects

IT3779
Smart Object Technologies

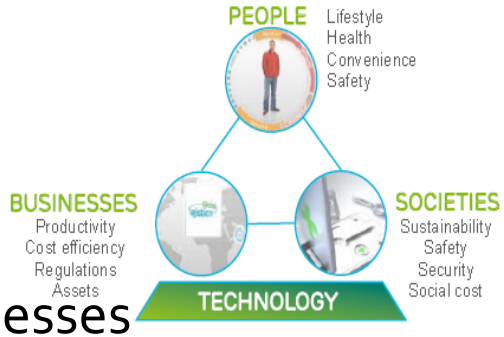
Outline

- Vision for Smart Objects/ Internet of Things (IoT)
- Current Status of Internet of Things
- Smart Object Architecture

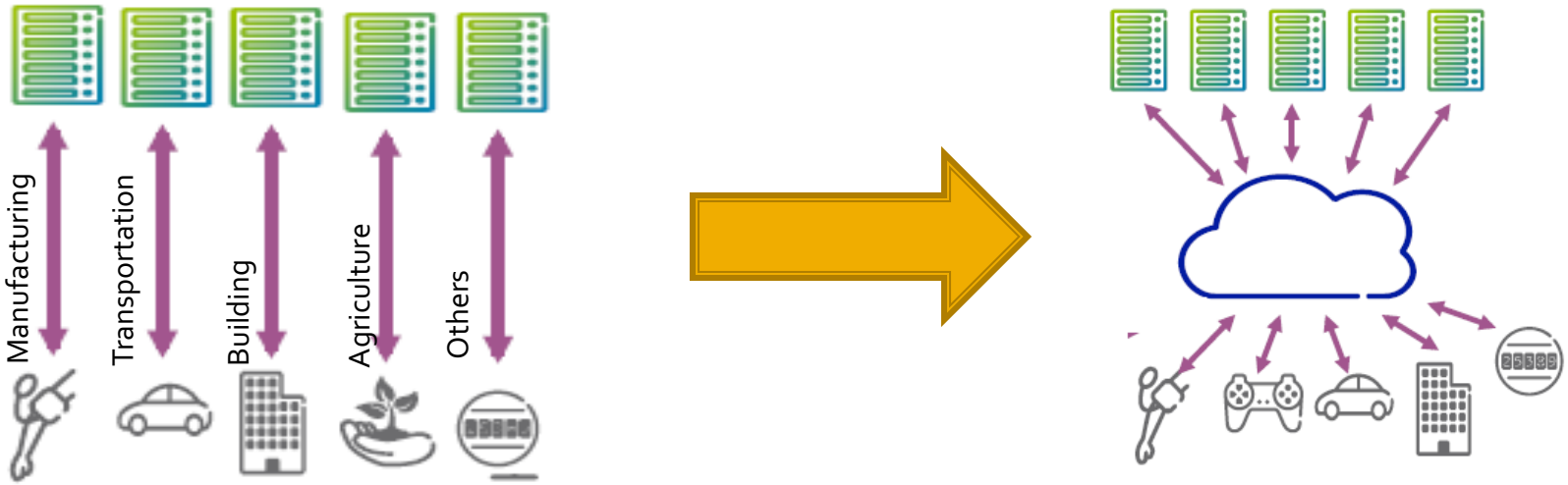
Vision

- The *vision* for Smart Objects or Internet of Things (IoT) is to connect billions of IoT seamlessly to the internet, interacting with humans and with each other

- Benefits of Internet of Things:
  - For people, improvements is lifestyle
  - For businesses exposed to an ever-increasing competitive business environment, new businesses opportunities, cost reduction, differentiation etc.
  - From a society perspective, saving energy, sustainability, efficiency and safety



Current Status of Internet of Things



- Today's IoT applications consists of vertically integrated end-to-end systems connected using a set of vendor specific Application Programmer's Interface (API) protocols
- The IoT vision is to transform it to a landscape where IoT applications operate seamlessly over a common platform

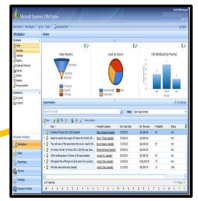
Typical Smart Object System

Vertical Applications
- Logistics
- Transport
- Buildings etc.

Info Processing & Storage
- Monitoring
- Decision making
- Search Engine

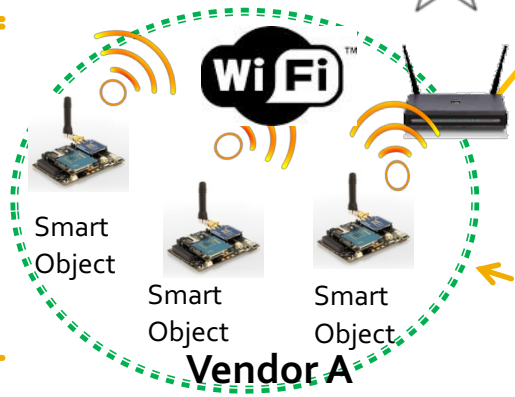
Communication Infrastructure

Sensing & Identification

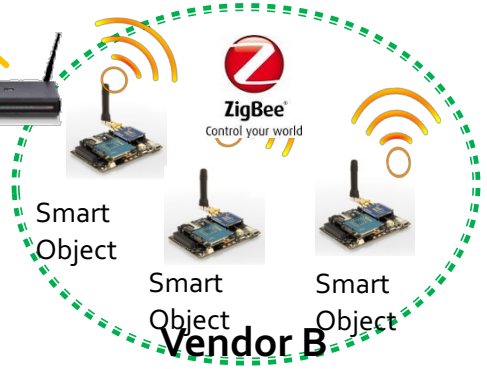


Cloud System

Gateways

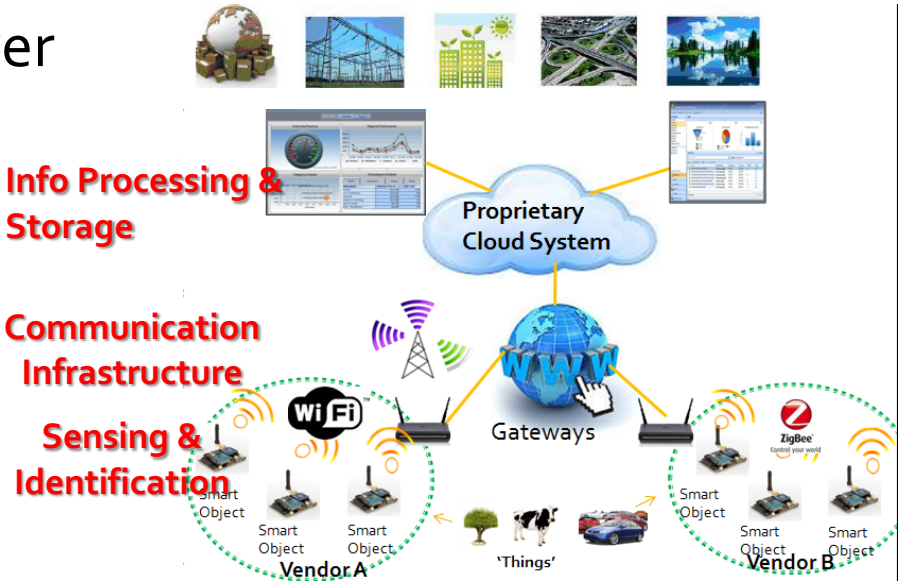


'Things'



Typical Smart Object System

- The **Sensor & Identification** Layer gathers data from surroundings using proprietary hardware solutions
- **Communication Infrastructure** facilitates communication to Internet via a Gateway using vendor specific protocols



- At the **Information Processing & Storage** Layer, sensor data is stored in Cloud System. The data is then used by applications for Visualization, Analytics and Monitoring services to end users for decision making

Sensing & Identification

- At the **Sensing & Identification** Layer, Smart Objects are deployed to monitor 'Things' in the physical environment

- What is a Smart Object?
  - A Smart Object is essentially, a tiny computer whose primary functions are:
    - Sense and collect info from the environment.
    - Communicate with other objects or systems over network
    - Report their own info and status
    - Receives information to update their own status

Sensing & Identification

- What does Smart Object 'sense'?

Vibration

Crack Detection

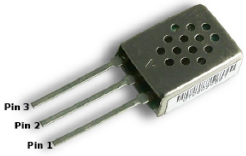


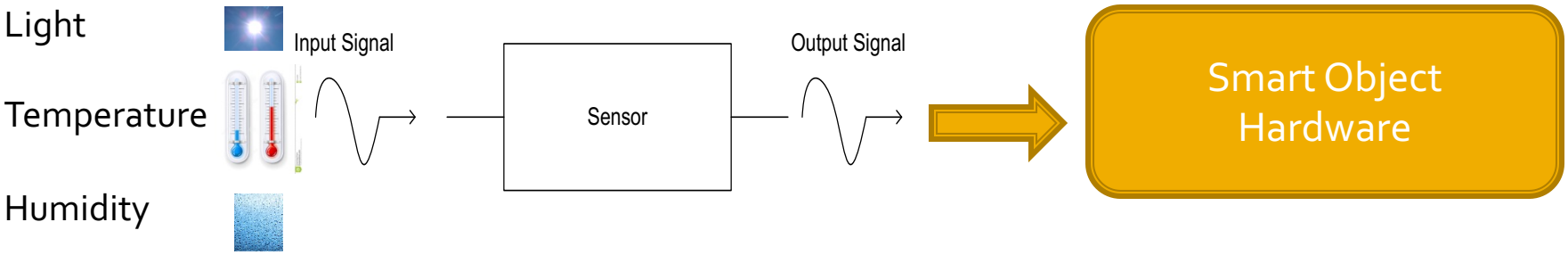
Figure 2: Image of the 808H5V5 sensor

Humidity

Temperature

… many more

Sensing & Identification

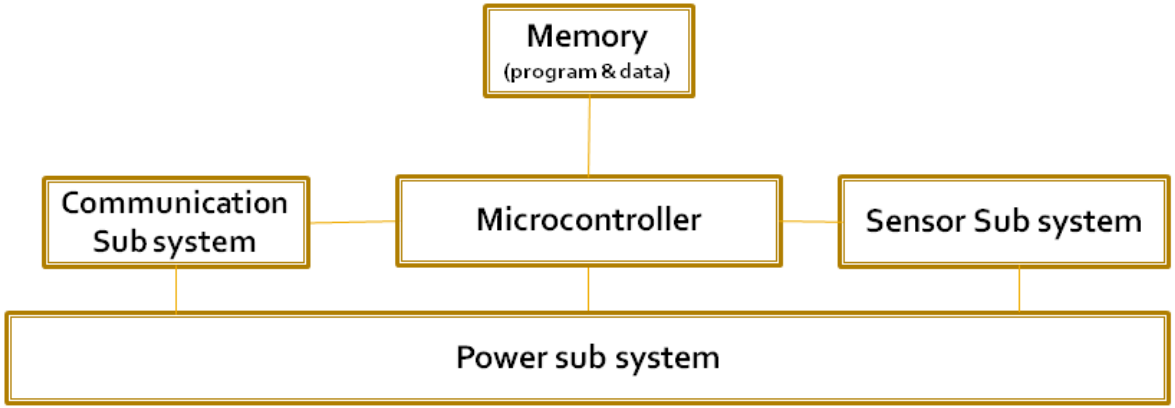


- Sensors acquires a **physical parameter** (e.g. temperature) and converts it into a signal suitable for processing (e.g. optical, electrical, mechanical) to be read by the Smart Object Hardware

Sensing & Identification

- **Smart Objects (SO) Hardware requirements**
  - Tiny in size : ubiquitous with limited memory & processing power
  - Low cost : deployed by thousands at low cost
  - Low energy consumption *:* battery need not be replaced or impossible to replace since many installations are deployed in remote locations

Sensing & Identification



- Smart Object Hardware comprises
  - **Microcontroller** acting as the brain for basic data processing & decision making
  - **Communication sub system** to communicate with other SO or Gateway
  - **Sensing sub system** to gather data from environment
  - **Power sub system** to power the Smart Object for months if not years

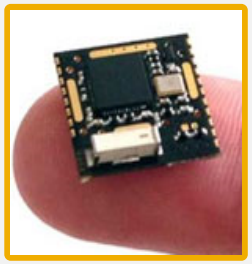
Sensing & Identification

Libelium Waspmote

An open source wireless sensor platform specially focused on the implementation of low consumption modes to allow the sensor nodes to be completely autonomous and battery powered

Raspberry Pi

A single-board computer developed in the UK by the Raspberry Pi Foundation. The Raspberry Pi is a credit-card sized computer that plugs into your TV and a keyboard



RFduino

A finger-tip sized wireless enabled microcontroller

Examples of Smart Object Hardware platforms

Sensing & Identification

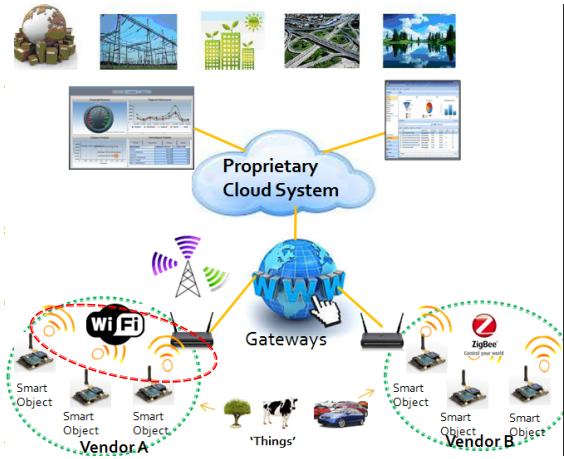
- **Smart Objects (SO) Operating System Software**
- An Smart Object Operating System is set of programs that manage computer hardware resources and provide common services such as
  - Scheduling
  - Memory Management
  - Communication Protocol
  - Resource Sharing
  - Support for Real-Time Applications

Sensing & Identification

- Smart Objects (SO) Operating System Software requirements
  - Have small memory footprint due to limited hardware memory
  - Be flexible i.e., only application-required services get loaded onto the system
- Examples of Operating System for Smart Objects

Communication Infrastructure

- **Smart Object Wireless Communication Technologies requirements:**
  - Low power : Power should last months if not years
  - Sufficient transmission distance : distance should be sufficient to cover a wide range
  - Typical low data rate : Data from smart objects do not consume large amount of data



Communication Infrastructure

- Common short range wireless communication technologies

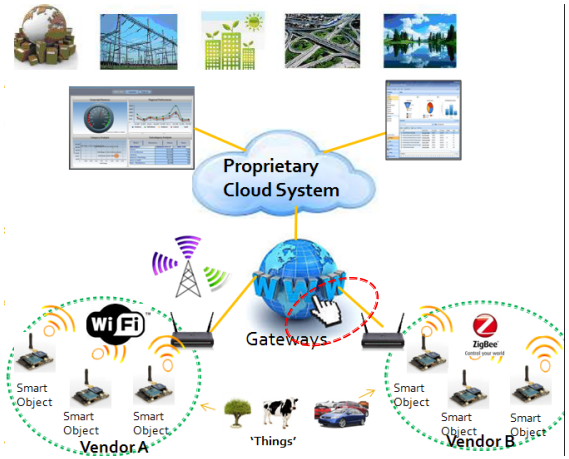
- Technology for exchange of data over a Wireless Local Area Network (WLAN)
- Present in most computers

- Wireless technology for short distances communication with high levels of security
- For communications between peripheral devices (e.g. mouse, keyboard, headsets etc.)

- Wireless technology specification for transmission protocols using small, low-power digital radios creating personal area networks (PANs)
- Used in short-range wireless data transfer for consumer and industrial systems (environmental monitoring etc.)

Communication Infrastructure

- A typical Smart Object local wireless network is connected to Cloud Platform over the Internet through Gateways



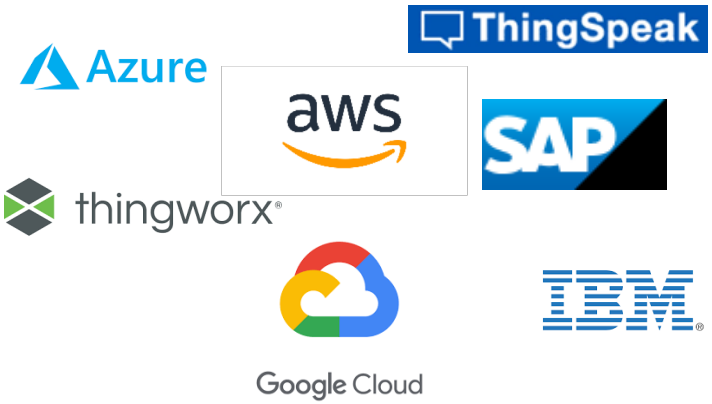
- The Gateway acts as a bridge that converts proprietary protocols used at the Smart Object network to HTTP-like protocols over the internet

Communication Infrastructure

- Evolving Communication standards
  - Existing lower power wireless communication technologies are proprietary & are not compatible with each other
  - The emerging trend is IP-enabled every Smart Object using IPv6
    - IP-based devices can be connected easily to other IP networks without using gateways.
    - IP networks allow the use of existing network infrastructure

Information Processing & Storage

- IP-enabled Smart Objects are able to access the internet
- Smart Objects data can then be stored, access & processed in the 'Cloud'
- There are many proprietary cloud services available

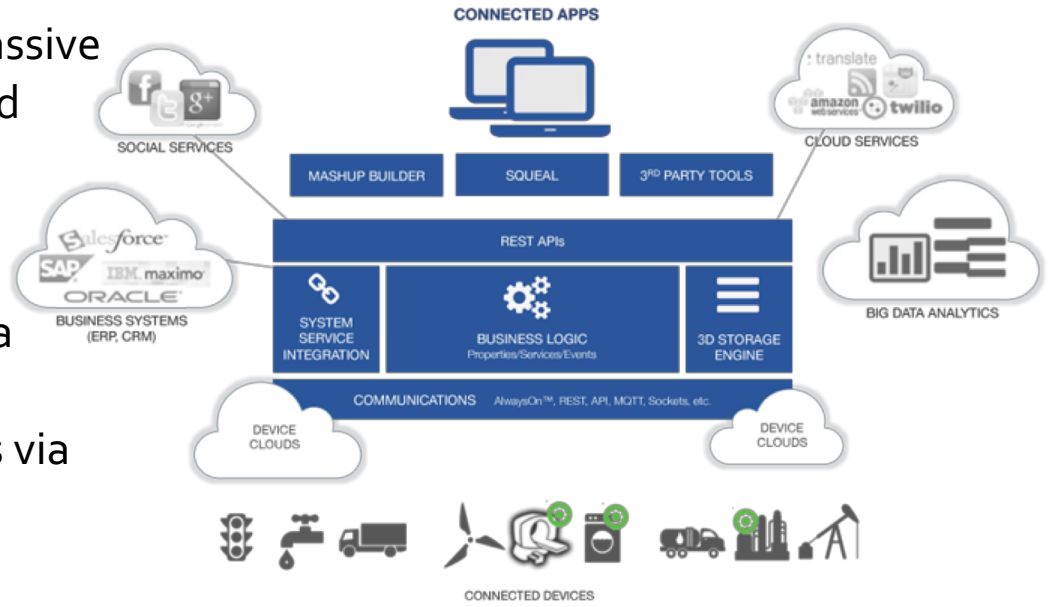


AWS - https://aws.amazon.com/iot/
Azure - https://azure.microsoft.com/en-us/services/iot-hub/
Google - https://cloud.google.com/solutions/iot/
IBM - https://www.ibm.com/cloud/internet-of-things/internet-of-things
SAP - https://cloudplatform.sap.com
ThingSpeak - https://www.thingspeak.com
Thingworx - https://www.ptc.com/en/products/iot

https://postscapes.com/internet-of-things-platforms

Information Processing & Storage

- Example of Cloud Platform Service: ThingWorx A PTC Business
  - Event-Driven Execution & Storage
    - Event-driven execution on massive amount of data stored in cloud allows useful actions to be performed
  - Search-based Intelligence
    - Online search on massive data
  - Flexible Connectivity
    - Connectivity to smart objects via various protocols & APIs

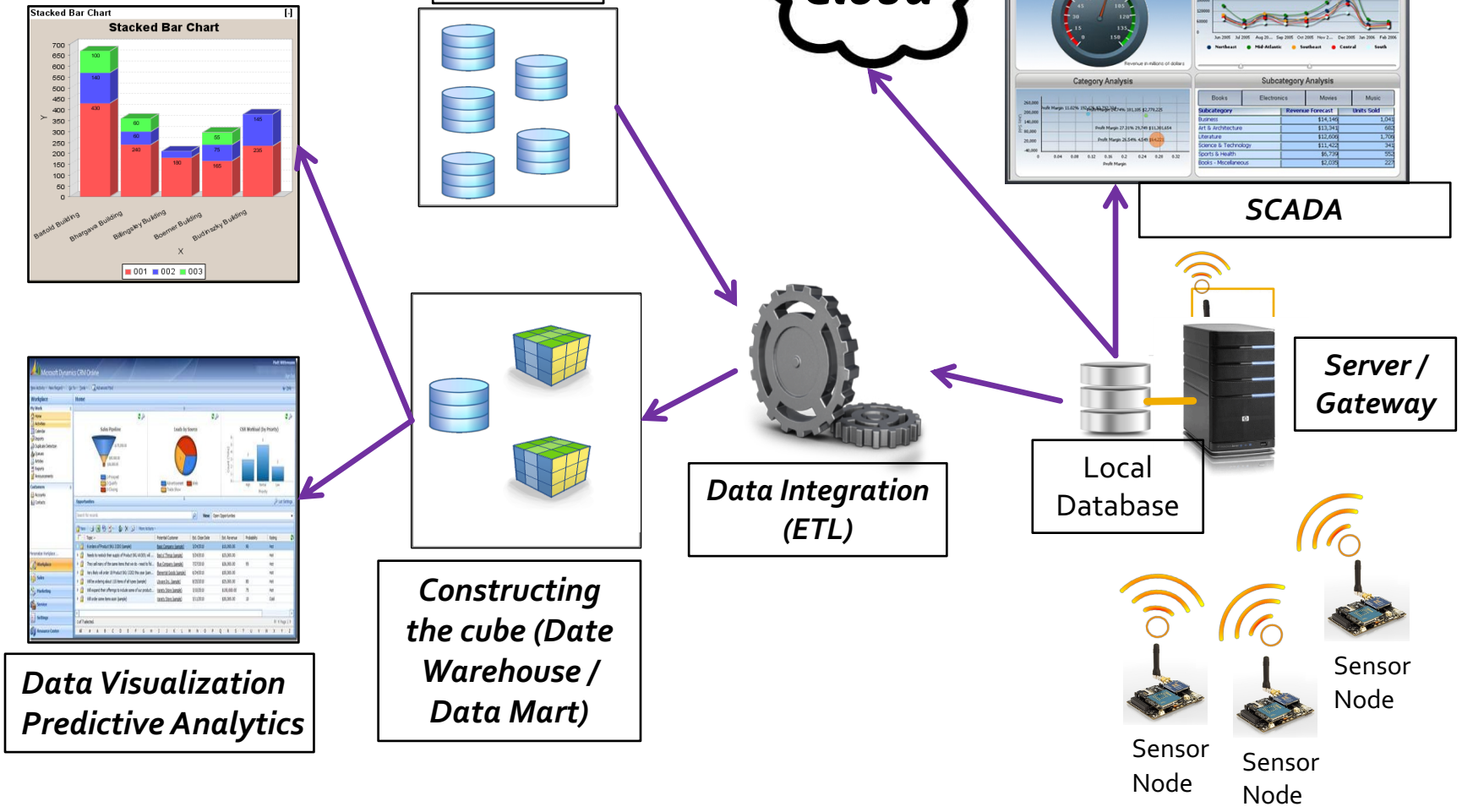


Information Processing & Storage

- Example of Cloud Platform Service: ETHERIOS

Click above to view video

Overview IoT System Architecture



Overview IoT System Architecture

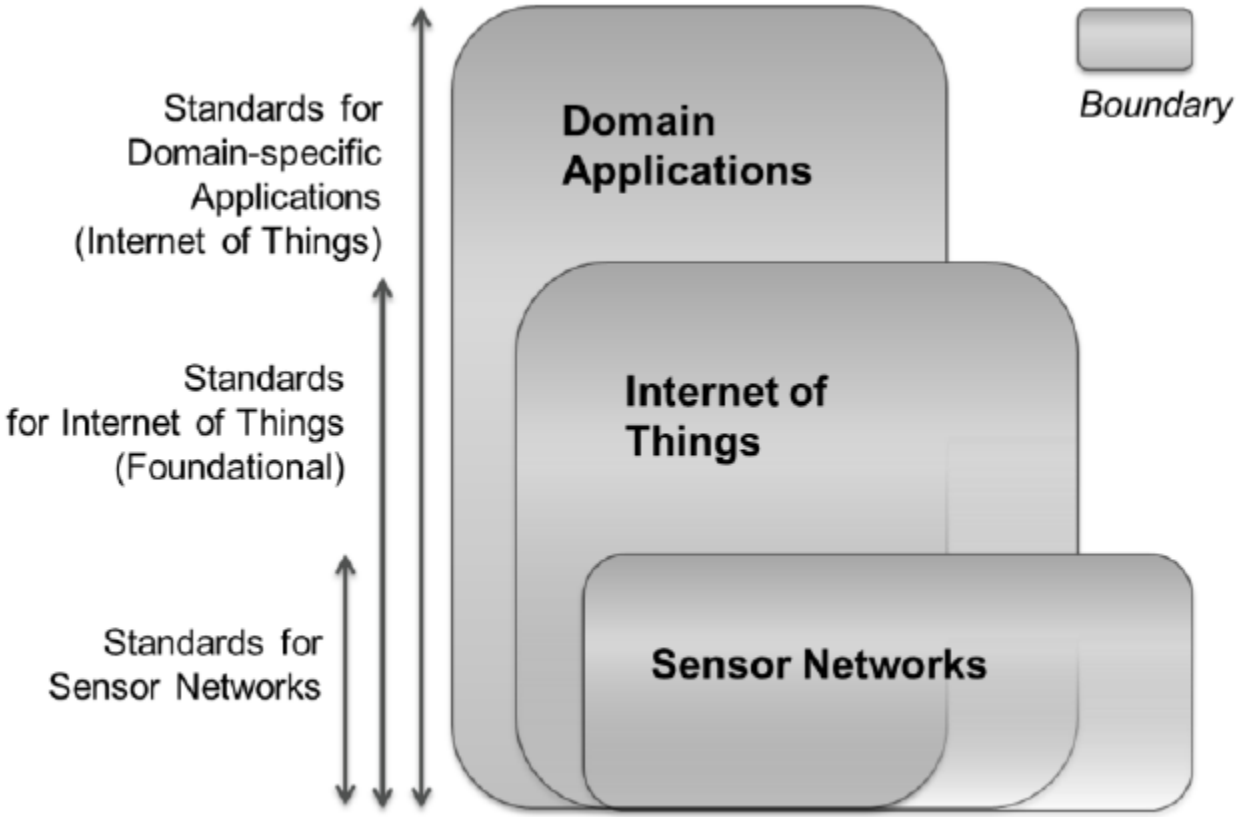
- Smart devices (sensors or actuators)
  - Sensors to collect physical parameters (e.g. temperature) and send to gateway. Actuators to control output (e.g. fan).
- Gateway/server collect and consolidate data to store to database.
- Database to store data collected.
- SCADA dashboard to display/monitor real time (current) data, alert and notification, and configuration and manual control (actuators).
- Cloud storage to store data for easy remote access/processing.
- External database contain other information for meaningful analysis.
- ETT/ETL (data integration) get data from operational system (current data), historical data and other information to create data cube (data warehouse / data mart).
- Data visualisation and analytic based on data from data cube

TR47 Technical Reference

Introduction

- Vision of a Smart Nation – focus areas include addressing the challenges of
  - Ageing population
  - Densely populated
  - Safe and secure data market place where individuals or companies can innovate to develop impactful applications and disruptive technologies
  - Etc.
- Use of IoT for Smart Nation initiatives – possible IoT applications and services include
  - Smart homes/buildings
  - Intelligent transportation and traffic
  - Tele-health
  - Environment monitoring systems
  - Etc.
- Suite of IoT standards for Smart Nation, Standards for Smart Nation are clustered into three groups as follows:
  1. Standards for sensor networks that aim to provide recommendations on the communication and application interface standards for the development and deployment of sensor network(s) for different environmental settings. The settings include, but are not limited to public areas and homes.
  2. **Standards for IoT (foundational) that provide information on basic building blocks which aid the development of cross-domain IoT systems for sharing information and the use of instrumentation. This TR is one of such standards.**
  3. Standards for domain-specific applications (Internet of Things) that focus on specific vertical domains such as healthcare and intelligent transportation.

Introduction



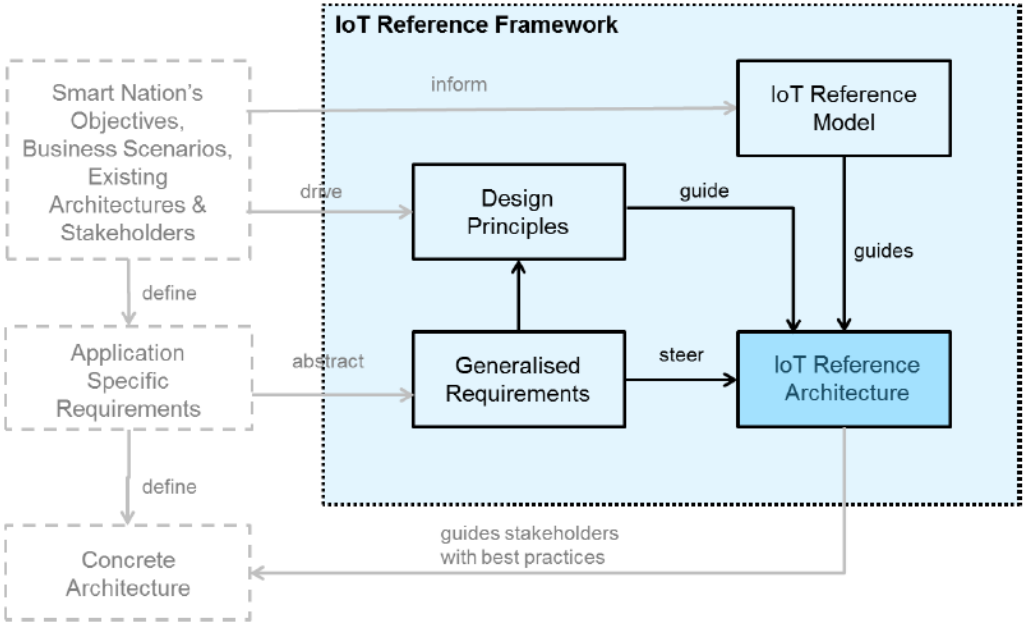
Suite of IoT standards for Smart Nation

Introduction

- This Technical Reference (TR) was prepared by the Internet of Things Reference Architecture Working Group (RAWG) of the Internet of Things Technical Committee (IoT TC), under the direction of the IT Standards Committee (ITSC). The ITSC endorsed the TR on 7 March 2016.
- Objectives of the TR:
  1. promote an open and common guiding architecture that facilitates the design and development of interoperable IoT systems to support Singapore's Smart Nation initiatives
  2. promote modularity in the components of IoT systems, so that they can be easily added to or removed from larger IoT systems
  3. guide the development of other IoT standards for Smart Nation
- The TR describes the following:
  - An IoT reference framework
  - A set of design principles
  - A set of general architectural requirements
  - An IoT reference model
  - An IoT reference architecture (IoT RA)

IoT reference framework

- Traditional software architecture development would consist of three stages:
  - Gathering the business objectives
  - Defining the application specific requirements
  - Developing the concrete architecture
- The IoT reference framework consists of four components:
  A. Design principles
  B. Generalised requirements
  C. IoT reference model
  D. IoT reference architecture



IoT Reference Framework

A) Design Principles

- The reference architecture presents IoT systems using abstraction levels that hide underlying complexities and heterogeneities associated with concrete architectures.
- In this IoT reference architecture, the following design principles have been considered:
  - Design for interoperability and reuse of deployed resources across domains;
  - Design for open service-oriented capabilities in solution development, deployment and support
  - Design for ensuring trust, security and privacy
  - Design for scalability, heterogeneity, adaptability and performance
  - Design for simplicity of management, integration and support

B) Generalised requirements and considerations for IoT RA

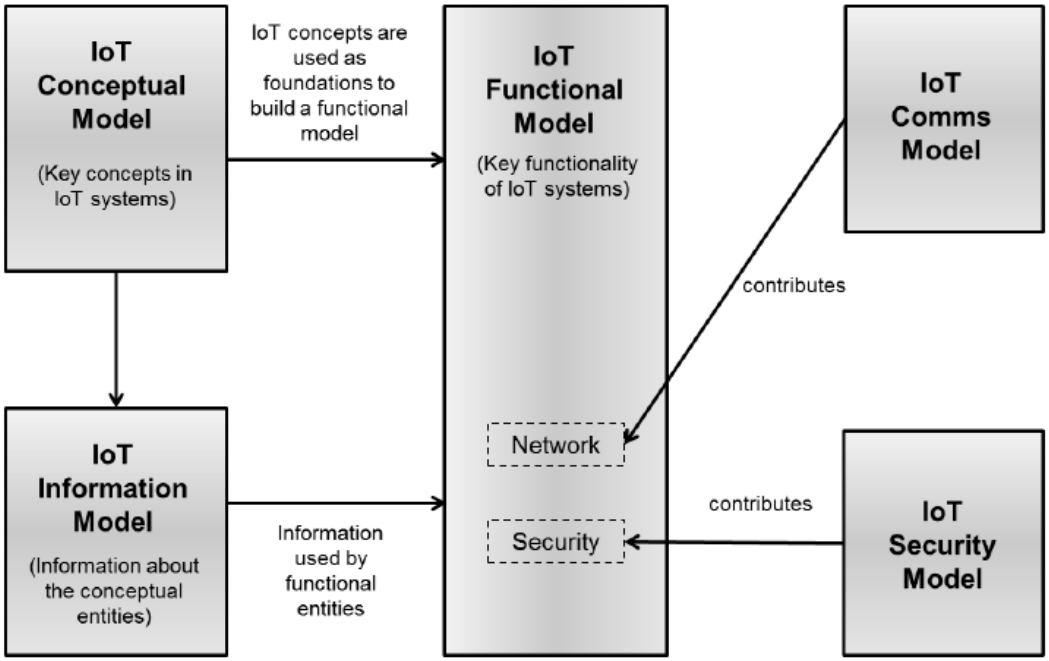
- Architectural requirements:
  - Domain agnostic – the IoT RA shall be designed to support services for use across domains
  - Technology independent – the IoT RA shall be independent of engineering and implementation technologies
  - Modular – the building blocks of IoT RA shall be loosely coupled
  - Collaboration with external systems – the IoT RA shall be designed such that a developed IoT system can collaborate or interoperate with external systems without imposing any architectural changes on either of them
  - Adaptable – the IoT RA shall be designed to support the progression of domain systems due to changes of requirements, system architectures, information flows and information types over time

B) Generalised requirements and considerations for IoT RA

- IoT applications considerations:
  - Sensor network device management
    - Autonomous functionality
    - Discoverability
    - Auto-configuration
  - Data, information and services
    - Content-awareness
    - Context-awareness
    - Location-awareness
    - Time-awareness
  - Interoperability
    - Standardised interfaces
    - Unique identification
  - Security, privacy, safety and regulatory
    - Security
    - Data privacy
    - Safety
    - Regulatory compliance
  - Operations and performance
    - Scalability
    - Reliability
    - Manageability
    - Availability
    - Timeliness

C) IoT Reference Model

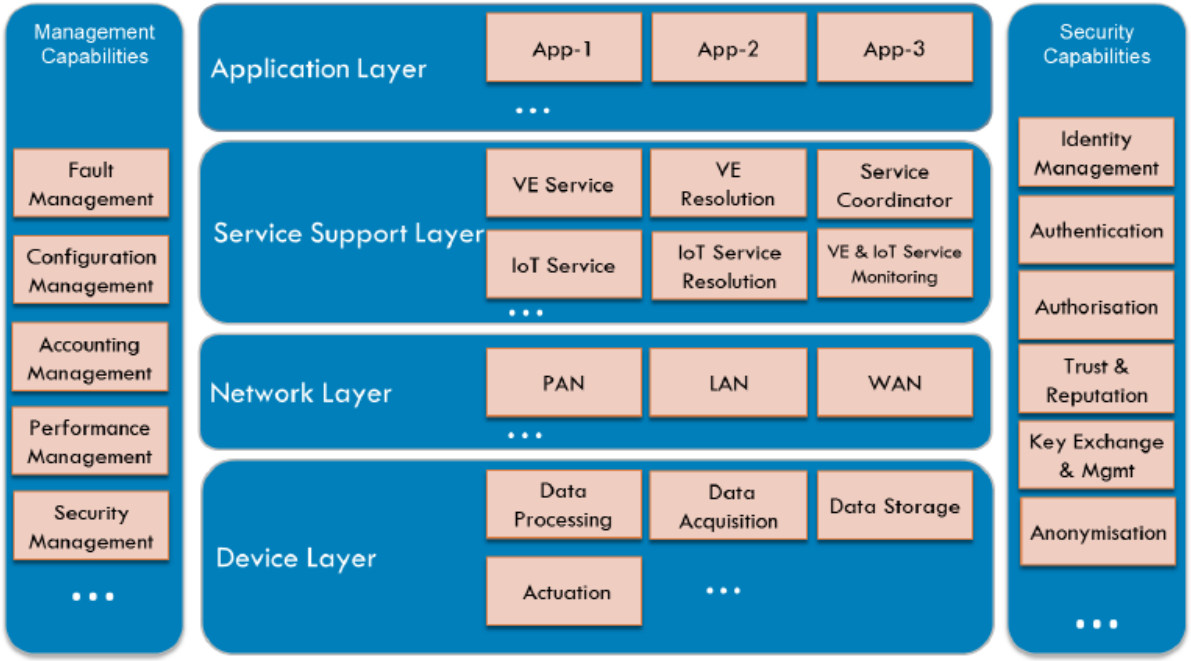
- According to OASIS (Organization for the Advancement of Structured Information Standards):
  - A reference model is an abstract framework for understanding significant relationships among the entities of some environment … A reference model consists of a minimal set of unifying concepts, axioms and relationships within a particular problem domain, and is independent of specific standards, technologies, implementations, or other concrete details.
- This TR is consistent with OASIS's definition, and comprises of five of sub-models:
  1) Conceptual model
  2) Information model
  3) Functional model
  4) Communication model
  5) Security model



IoT Reference Model

D) IoT Reference Architecture

- Higher abstraction of system architectures which have similar characteristics when they are used to develop actual systems
- Used to guide the creation of concrete architectures for actual systems
- Derived from the IoT Reference Model
- Described from a functional and an interface viewpoint



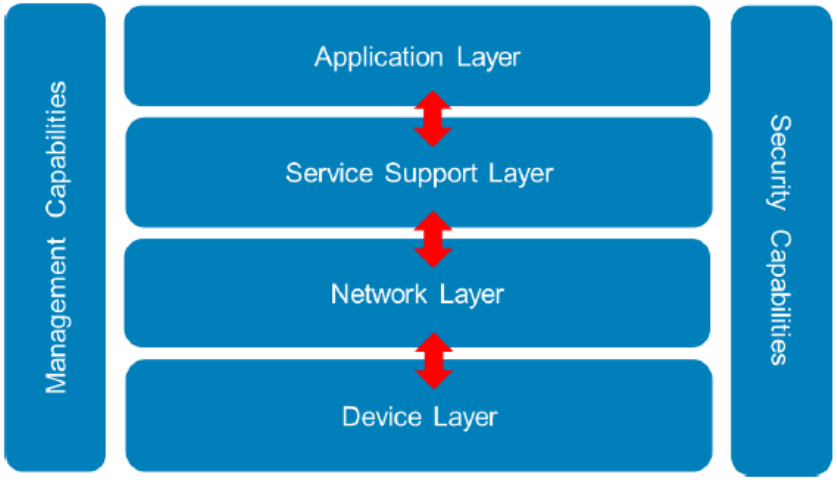
IoT RA Functional View

D) IoT RA – Functional View

- Functionalities provided by each of the layers
  - Device layer
    - As mentioned in IoT Functional Model
  - Network layer
    - Services delivered by the network layer are referred to as network resource services
  - Service support layer
    - Consists of generic services used by IoT applications
    - Contains functions that facilitate the interactions between users and “things”
  - Application layer
    - Contains specific IoT applications
  - Management capabilities
    - Configuration management – provides functionality for retrieving and setting various configurations of the system
    - Fault management – provides functionality to handle, monitor and retrieve faults
    - Accounting management – concerns with tracking network utilisation information and contributes to the overall security and performance of the system
    - Performance management – concerns with the health and performance of the overall system
    - Security management – refers to the ability of the IoT management systems to manage the overall security of the operating environment, including the verification of security configurations, monitoring of security operation of all the critical services, services related to security capabilities (as mentioned in IoT Functional Model) as well as detecting and coordinating responses to security events
  - Security capabilities
    - As mentioned in IoT Functional Model

D) IoT RA – Interface View

- Standardise the interfaces between the layered structures in IoT systems and their applications to achieve interoperability
- Interfaces refer to the communication protocols or data handling methods for each entity in the layers



IoT RA high-level interfaces

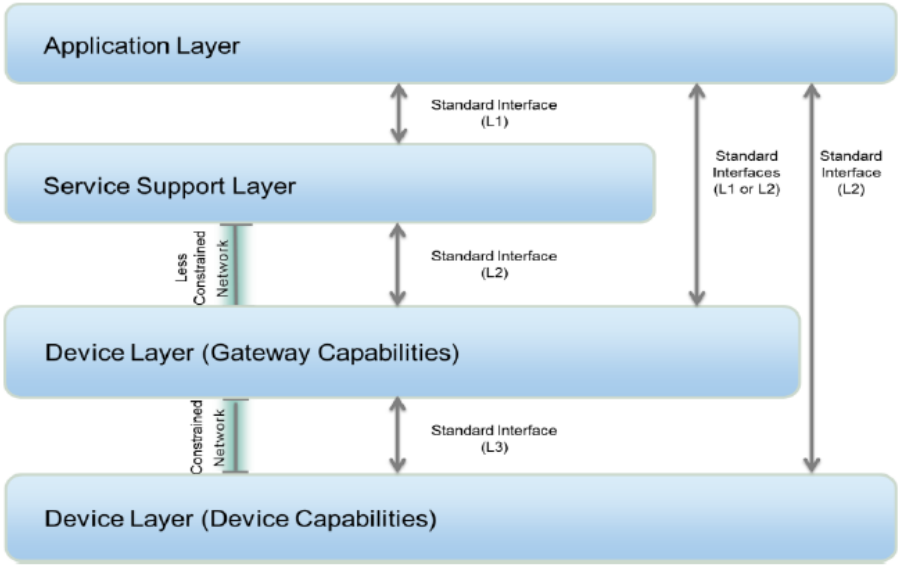
- Two main flows of information in an IoT system
  - from devices that produce information such as sensors and tags, the information follows a context-enrichment process until it reaches the consumer application or part of the larger system
  - from the application or part of a larger system, the information follows a context-reduction process until it reaches the consumer devices (e.g. actuators)

D) IoT RA – Interface View

- To develop application-specific IoT architecture, shall define
  - Interfaces for each and every module inside the service support layer, application layer and network layer
  - Interfaces between the service support layer and application layer
  - Interfaces between the service support layer and network layer
  - Interfaces between the network layer and the device layer
- IoT RA standard interfaces
  - Categorises the interfaces into three standard interfaces L1, L2, and L3
  - Omits the network layer, management capabilities and security capabilities
  - Focuses on the data and control interfaces between the application layer, service support layer and device layer
  - Device layer is further broken down into gateway capabilities and device capabilities as they might have different interface requirements.

D) IoT RA – Interface View

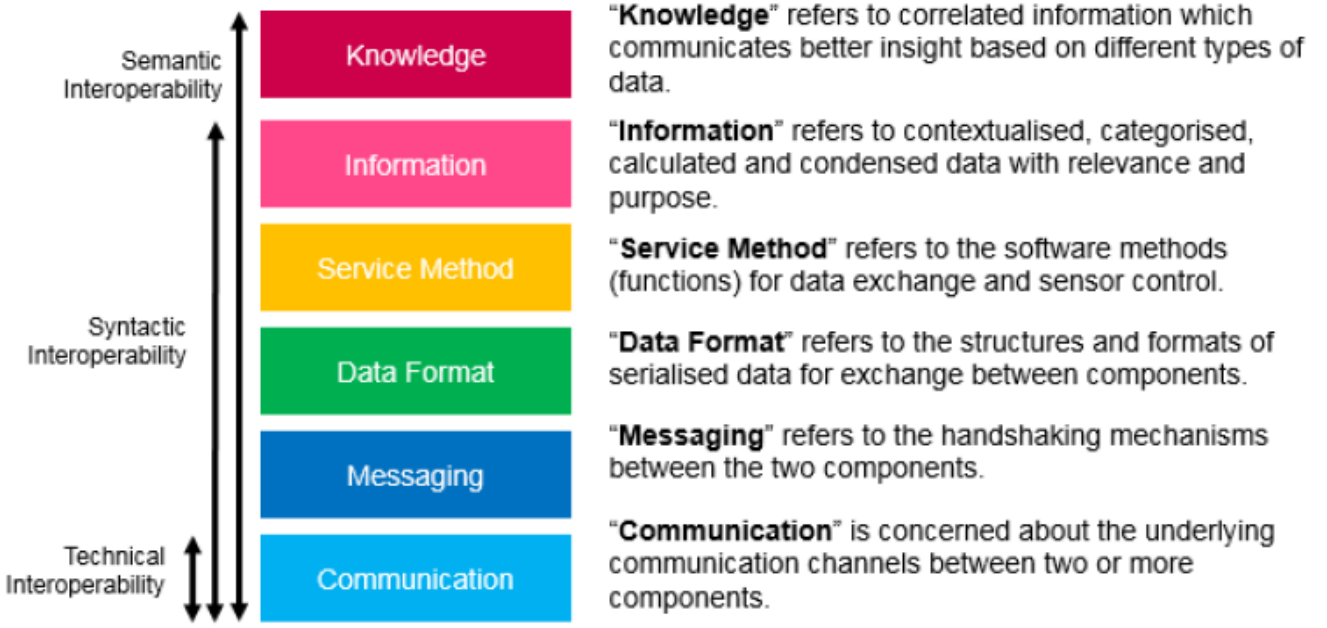
- Standard interface (L1)
  - Between service support layer and application layer
  - It is also possible for a “heavy” gateway from the device layer (gateway capabilities) to interface directly with the application layer using L1
- Standard interface (L2)
  - Between device layer (gateway capabilities) and service support layer.
  - It is also possible for an application to implement L2 to interface directly with a gateway or sensor node residing on the device layer (gateway capabilities) or device layer (device capabilities) respectively.
- Standard interface (L3)
  - Between device layer (gateway capabilities) and device layer (device capabilities).



IoT RA Interface View

D) IoT RA – Interface View

- Six aspects of standard interfaces
  - An interface is a means to exchange messages between two endpoints, and standardising the interfaces helps to improve interoperability
  - Standard interfaces are broken down into six aspects
    - Knowledge
    - Information
    - Service Method
    - Data Format
    - Messaging
    - Communication



Aspects of a data and control interface

D) IoT RA – Interface View

- These different aspects help to achieve interoperability between systems at three different levels
  - Technical interoperability – concerns with the physical transfer of data and is typically related to communication
  - Syntactic interoperability – focuses on data structures and formats, where data transmitted can be constructed into different message types with different content.
  - Semantic interoperability – defines the meaning and the use of the data received as well as enables systems to interpret the data automatically in a meaningful manner.

End