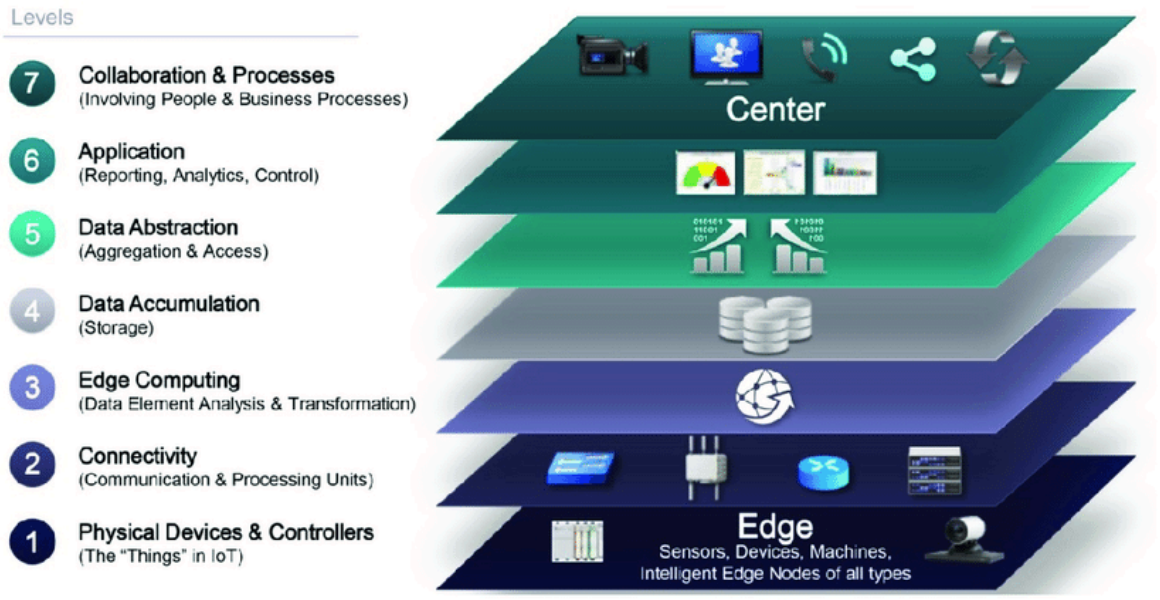
IoT REFERENCE MODEL

Referenced from Cisco Document

The IoT Reference Model is a conceptual framework that helps standardize how Internet of Things (IoT) systems are structured, designed, and implemented. It provides different layers, each representing a specific function, to ensure that IoT solutions are interoperable, scalable, and secure. The model organizes the complex components of IoT into manageable layers that guide the development of IoT solutions.



1. Device Layer (Physical Devices and Controllers)

Function: This layer represents the "things" in IoT—various physical devices, sensors, actuators, and controllers that gather data from the environment or perform actions. Devices are diverse, and there are no rules about size, location, form factor, or origin. Some devices will be the size of a silicon chip. Some will be as large as vehicles. The IoT must support the entire range. Dozens or hundreds of equipment manufacturers will produce IoT devices. To simplify compatibility and support

Role: These devices either sense the environment (collect data such as temperature, humidity, or motion) or control physical processes (like turning on lights or adjusting a thermostat).

Components: Sensors, actuators, RFID tags, smart devices, etc.

Example: A temperature sensor embedded in a smart thermostat continuously measures the temperature and sends readings as MQTT messages to an MQTT broker.

2. Network Layer (Connectivity)

Function: This layer facilitates communication between IoT devices and the other layers of the system. This includes transmissions:

● Between devices (Level 1) and the network

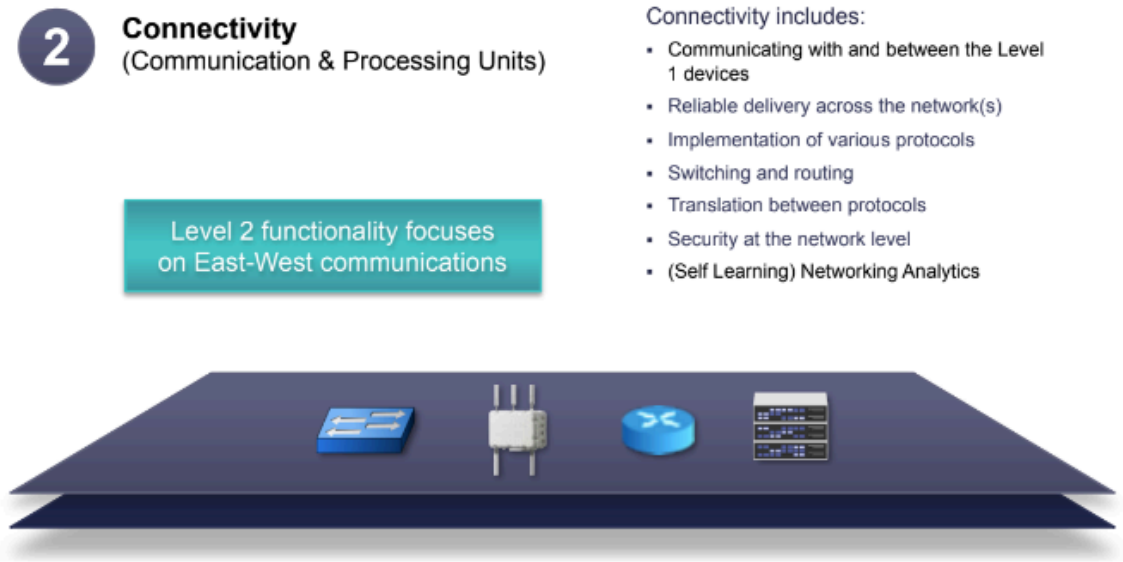
● Across networks (east-west)

● Between the network (Level 2) and low-level information processing occurring at Level 3

Role: The network layer transfers data from the physical devices to the cloud or other network destinations for processing.

Components: Routers, gateways, network protocols, and communication technologies (Wi-Fi, ZigBee, Bluetooth, LoRa, 5G).

Example: The smart thermostat sends temperature data via TCP/IP to the MQTT broker over a Wi-Fi or cellular network.



3. Processing Layer (Edge Computing)

Function: This layer processes data at the edge of the network, meaning close to the source of data generation. Level 3 processing can encompass many examples, such as:

● Evaluation: Evaluating data for criteria as to whether it should be processed at a higher level

● Formatting: Reformatting data for consistent higher-level processing

● Expanding/decoding: Handling cryptic data with additional context (such as the origin)

● Distillation/reduction: Reducing and/or summarizing data to minimize the impact of data and traffic on the network and higher-level processing systems

● Assessment: Determining whether data represents a threshold or alert; this could include redirecting data to additional destinations

Role: By processing data at the edge, latency is reduced, which is critical for time-sensitive applications like autonomous driving or industrial automation.

Components: Edge servers, local analytics software, or on-device computing.

Example: An edge gateway subscribes to MQTT topics related to temperature data, processes it locally, and only forwards relevant information to the cloud for further analysis.

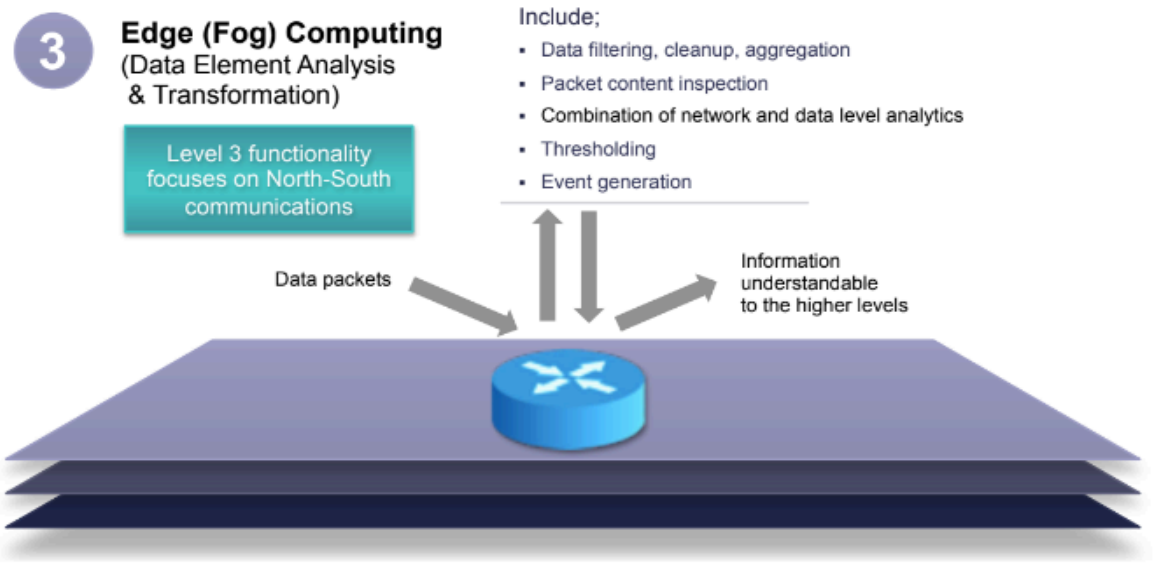
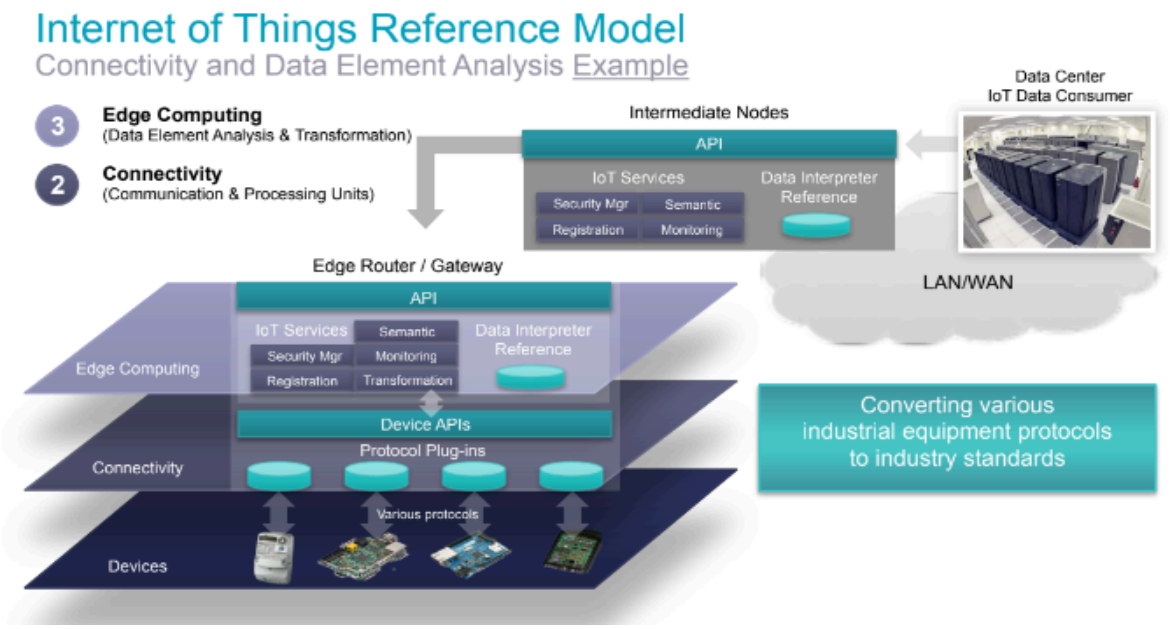


Figure 5. Level 2 and 3 Connectivity and Data Element Analysis Example



4. Data Layer (Data Accumulation)

Function: This layer is responsible for storing the data gathered from IoT devices. Level 4 determines:

● If data is of interest to higher levels: If so, Level 4 processing is the first level that is configured to serve the specific needs of a higher level.

● If data must be persisted: Should data be kept on disk in a non-volatile state or accumulated in memory for short-term use?

● The type of storage needed: Does persistency require a file system, big data system, or relational database?

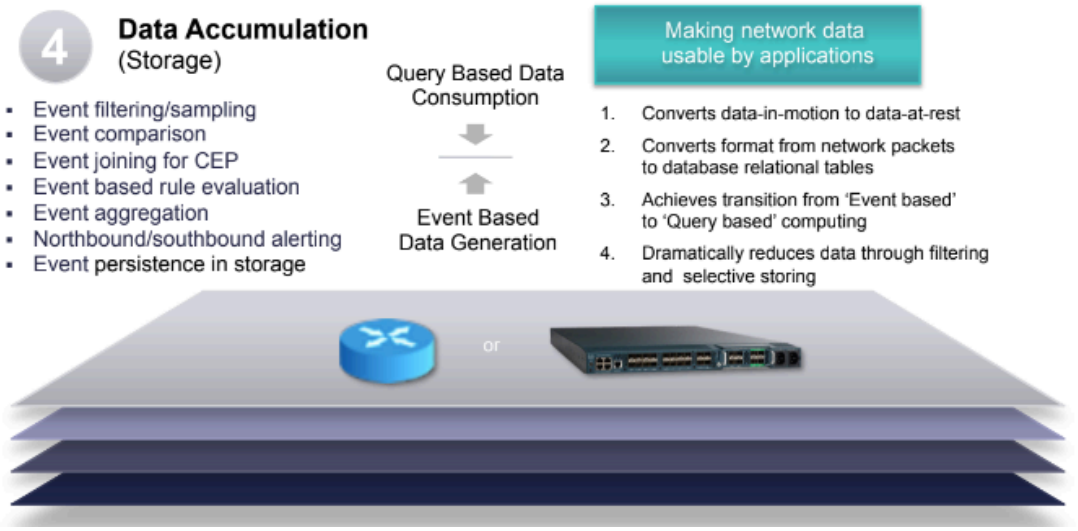
● If data is organized properly: Is the data appropriately organized for the required storage system?

● If data must be recombined or recomputed: Data might be combined, recomputed, or aggregated with previously stored information, some of which may have come from non-IoT sources.

Role: It accumulates both real-time and historical data to be analyzed later or to be directly used for real-time decision-making.

Components: Databases, data lakes, storage systems.

Example: The MQTT broker collects temperature readings and forwards them to a cloud database for storage and analysis.



5. Service Layer (Data Abstraction)

Function: This layer abstracts the raw data into meaningful information. With multiple devices generating data, there are many reasons why this data may not land in the same data

storage:

● There might be too much data to put in one place.

● Moving data into a database might consume too much processing power, so that retrieving it must be separated from the data generation process. This is done today with online transaction processing (OLTP) databases and data warehouses.

● Devices might be geographically separated, and processing is optimized locally.

● Levels 3 and 4 might separate “continuous streams of raw data” from “data that represents an event.” Data storage for streaming data may be a big data system, such as Hadoop. Storage for event data may be a relational database management system (RDBMS) with faster query times.

● Different kinds of data processing might be required. For example, in-store processing will focus on different things than across-all-stores summary processing.

For these reasons, the data abstraction level must process many different things. These include:

● Reconciling multiple data formats from different sources

● Assuring consistent semantics of data across sources

● Confirming that data is complete to the higher-level application

● Consolidating data into one place (with ETL, ELT, or data replication) or providing access to multiple data stores through data virtualization

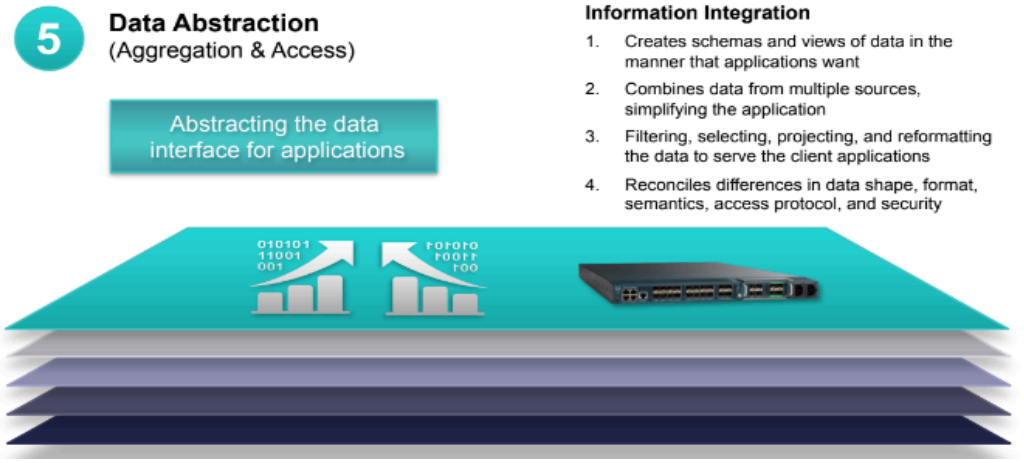
● Protecting data with appropriate authentication and authorization

● Normalizing or denormalizing and indexing data to provide fast application access

Role: It transforms the raw data into structured and usable information. This layer provides services that applications can use to access and interpret the data.

Components: Middleware, APIs, and data abstraction tools.

Example: The system organizes incoming MQTT messages by "topics" such as home/livingroom/temperature, making it easier for applications to access and interpret the data.



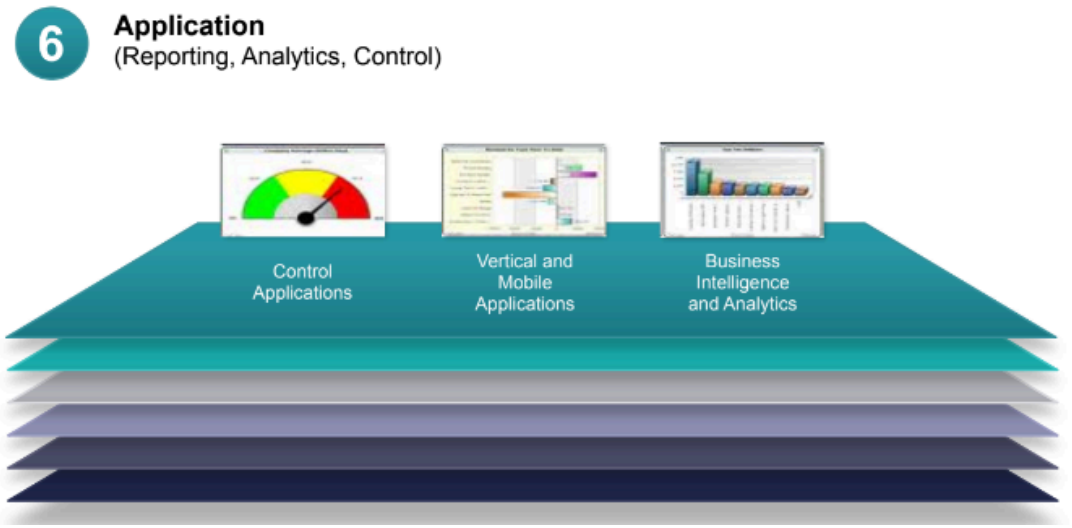
6. Application Layer (Application)

Function: This is where the IoT-generated data is utilized for specific applications.

Role: It provides the end-user functionality where the actual use cases of IoT come into play, such as smart homes, smart grids, or connected vehicles.

Components: Smart city applications, industrial control systems, consumer applications like home automation or wearable health tech.

Example: A home automation app displays the current temperature on the user's phone and allows them to adjust the thermostat remotely by publishing an MQTT message with a new temperature setting.



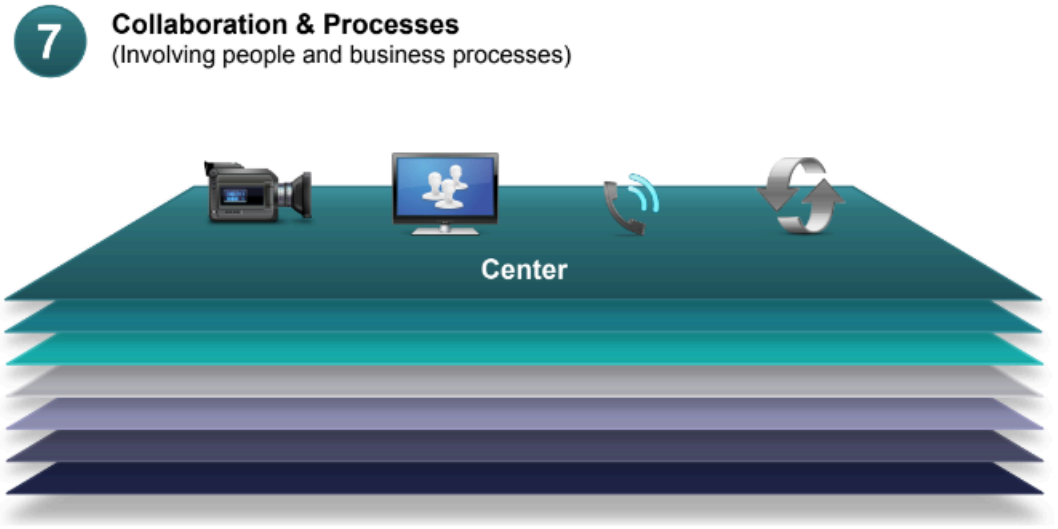
7. Collaboration Layer (Business and Ecosystem)

Function: This layer represents the collaboration between IoT systems and the external ecosystem. People must be able to communicate and collaborate, sometimes using the traditional Internet, to make the IoT useful.

Role: It ensures that the IoT system aligns with business goals, regulatory standards, and broader ecosystem collaboration.

Components: Business models, partner integrations, regulatory compliance tools.

Example: A company provides an IoT platform for smart homes using MQTT for device-to-cloud communication. The platform integrates with various partners for device compatibility and adheres to regulatory standards for data privacy.



8. Security

Function: This layer provides security across all other layers of the IoT model. For the purpose of the IoT Reference Model, security measures must:

- Secure each device or system
- Provide security for all processes at each level
- Secure movement and communication between each level, whether north- or south-bound

Role: The security layer ensures data privacy, device authentication, and secure communication between devices and networks.

Components: Encryption, authentication mechanisms, threat detection systems.

Example: The MQTT broker uses TLS to ensure that communication between the smart thermostat and the cloud is encrypted. Additionally, devices authenticate with the broker using certificates to ensure only authorized devices can publish or subscribe to topics.

Why IoT Reference Model?

Interoperability: Ensures that devices and systems can work together, regardless of vendor or platform.

Scalability: Supports systems from small-scale setups to vast, global networks.

Security: A dedicated security layer ensures that data and devices are protected from unauthorized access.

Modularity: The layer-based approach allows developers to focus on specific areas without needing to manage the entire system.


ON
IOT
(INTERNET OF THINGS)