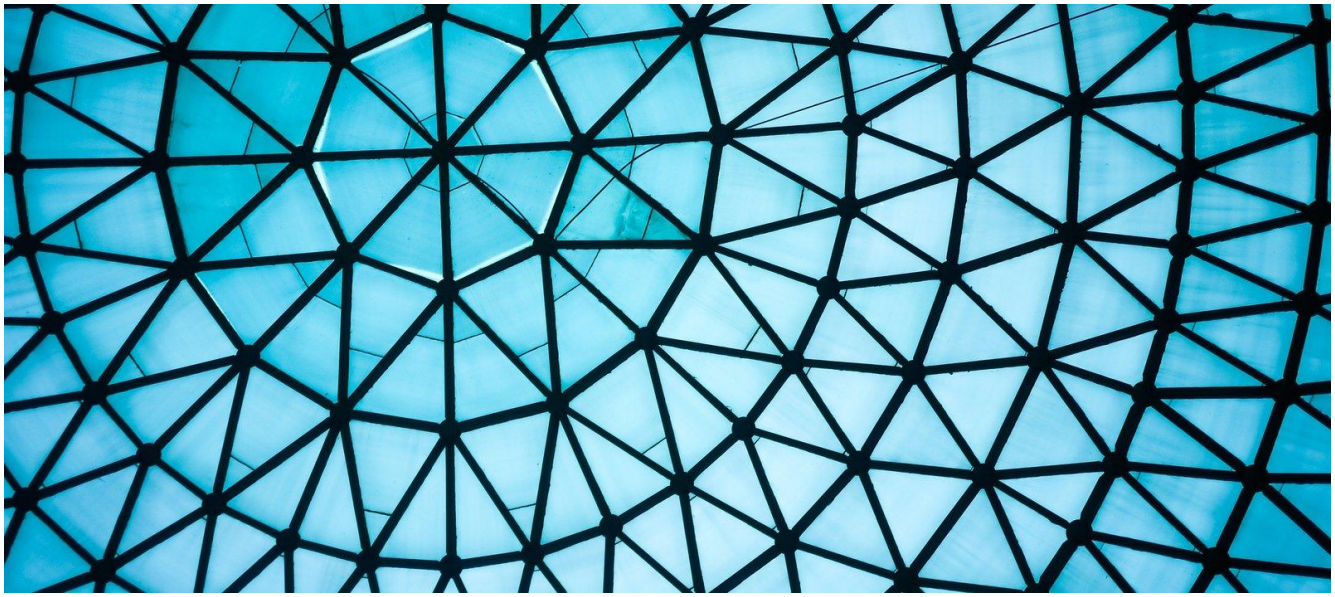
https://www.altexsoft.com/blog/iot-architecture-layers-components/



IoT Architecture: the Pathway from Physical Signals to Business Decisions

IoT solutions have become a regular part of our lives. From the smartwatch on your wrist to industrial enterprises, connected devices are everywhere. Having *thing*, work for us is no longer sci-fi fantasy.

You tap the screen of your smartphone or say a word, and get immediate results. A door automatically opens, a coffee machine starts grinding beans to make a perfect cup of espresso while you receive analytical reports based on fresh data from sensors miles away.

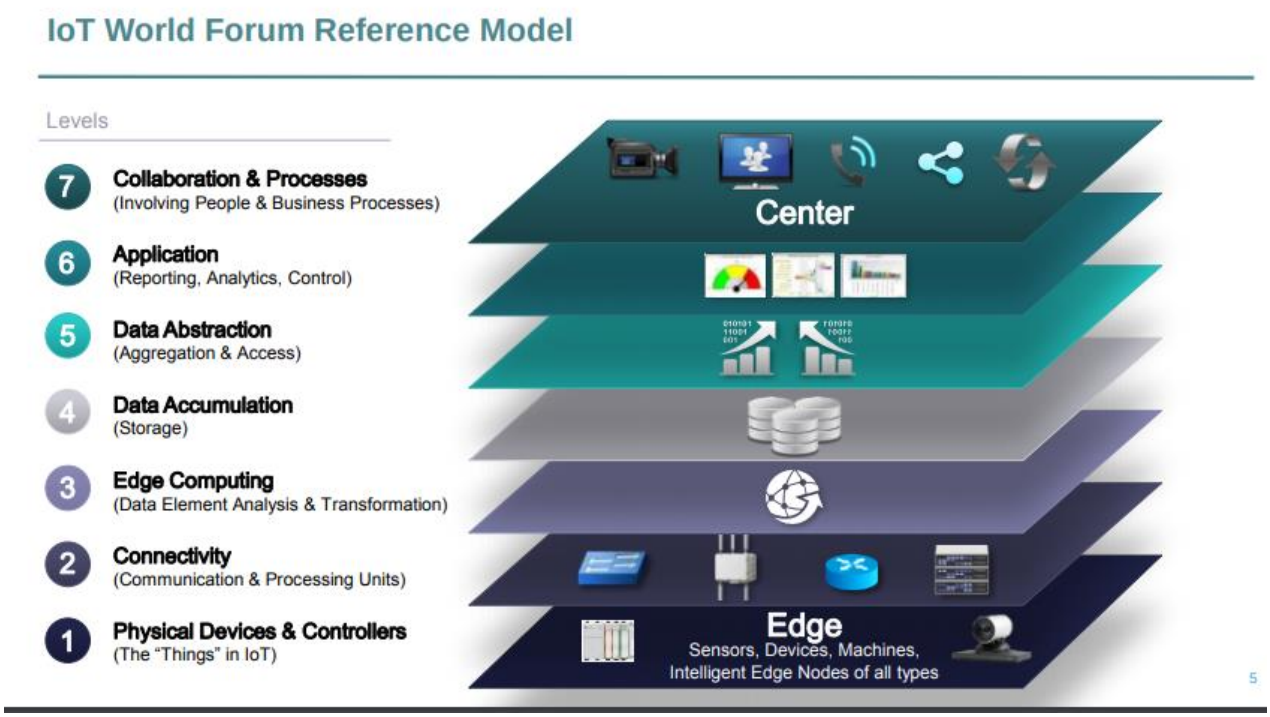
But between your command and tasks fulfilled, there lies a large and mostly invisible infrastructure, that involves multiple elements and interactions. This article describes IoT — the Internet of Things — through its architecture, layer to layer. Let's peek behind the curtain to see how everyday magic works.

Major IoT building blocks and layers

Before we go any further, it's worth pointing out that there is no single, agreed-upon IoT architecture. It varies in complexity and number of architectural layers depending on a particular business task.

For example, the Reference Model introduced in 2014 by Cisco, IBM, and Intel at the 2014 IoT World Forum has as many as seven layers. According to an official press release by Cisco forum host, the

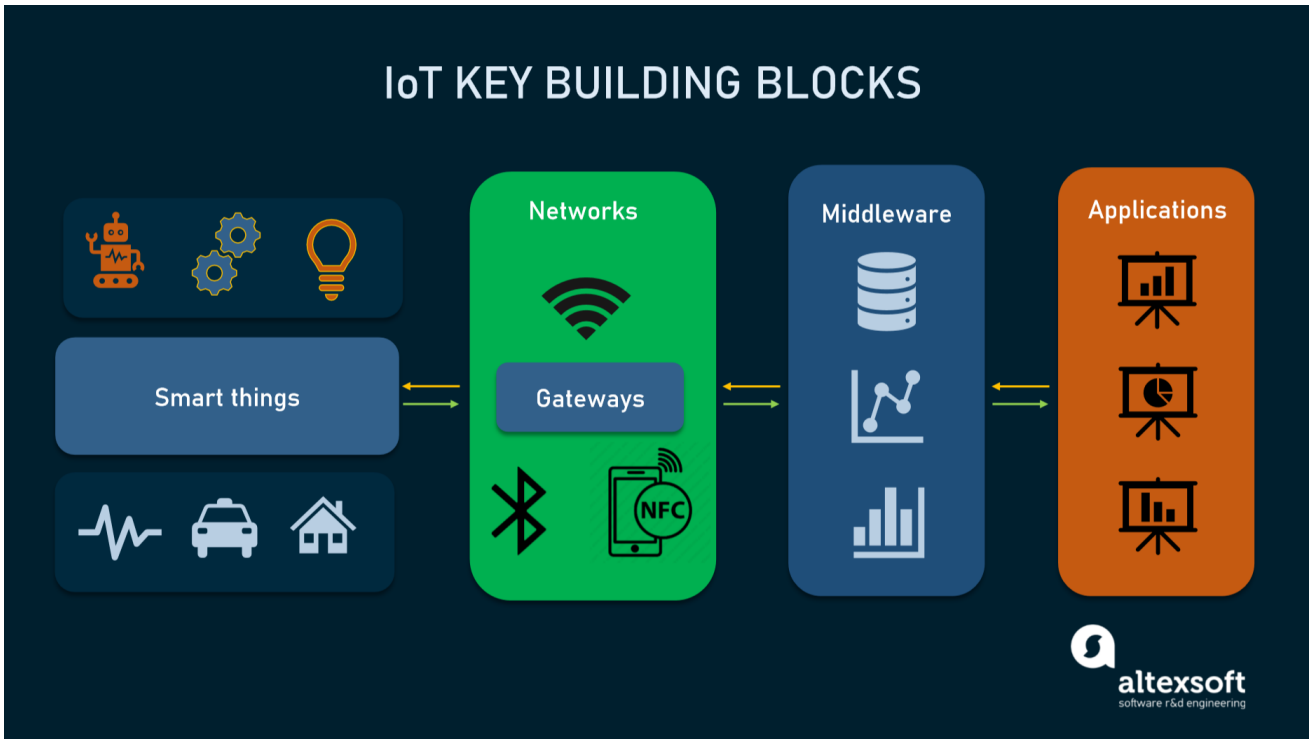
architecture aims to *"help educate CIOs, IT departments, and developers on deployment of IoT projects, and accelerate the adoption of IoT."*



The standardized architectural model proposed by IoT industry leaders. Source: Internet of Things World Forum

But no matter the use case and number of layers, the key building blocks of any IoT structure are always the same, namely:

- **smart things**;
- **networks** and **gateways** enabling low-power devices (which is often the case in IoT) to enter the big Internet;
- the **middleware or IoT platforms** providing data storage spaces and advanced computing engines along with analytical capabilities; and
- **applications,** allowing end users to benefit from IoT and manipulate the physical world.



The skeleton of an IoT system.

These elements make up the backbone of any IoT system upon which effective, multi-layered architecture can be developed. Most commonly, these layers are:

- the **perception layer** hosting smart things;
- the **connectivity or transport layer** transferring data from the physical layer to the cloud and vice versa via networks and gateways;
- the **processing layer** employing IoT platforms to accumulate and manage all data streams; and
- the **application layer** delivering solutions like analytics, reporting, and device control to end users.

Besides the most essential components, the article also describes three additional layers:

- the **edge or fog computing layer** performing data preprocessing close to the edge, where IoT things collect new information. Typically, edgy computing occurs on gateways;
- the **business layer** where businesses make decisions based on the data; and
- the **security layer** encompassing all other layers.

Often viewed as optional, these extra components nonetheless make an IoT project neatly fit modern business needs.

Perception layer: converting analog signals into digital data and vice versa

The initial stage of any IoT system embraces a wide range of “things” or endpoint devices that act as a bridge between the real and digital worlds. They vary in form and size, from tiny silicon chips to large vehicles. By their functions, IoT things can be divided into the following large groups.

Sensors such as probes, gauges, meters, and others. They collect physical parameters like temperature or humidity, turn them into electrical signals, and send them to the IoT system. IoT sensors are typically small and consume little power.

Actuators, translating electrical signals from the IoT system into physical actions. Actuators are used in motor controllers, lasers, robotic arms.

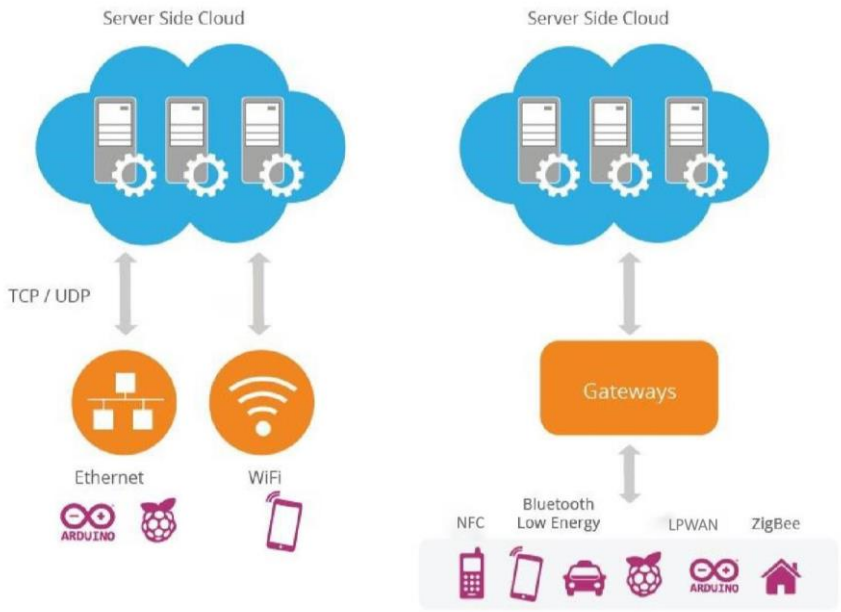
Machines and devices connected to sensors and actuators or having them as integral parts.

It’s important to note that the architecture puts no restriction on the scope of its components or their location. The edge-side layer can include just a few “things” physically placed in one room or myriads of sensors and devices distributed across the world.

Connectivity layer: enabling data transmission

The second level is in charge of all communications across devices, networks, and cloud services that make up the IoT infrastructure. The connectivity between the physical layer and the cloud is achieved in two ways:

- directly, using TCP or UDP/IP stack;
- via gateways — hardware or software modules performing translation between different protocols as well as encryption and decryption of IoT data.



Two key models of connectivity between physical and cloud levels in IoT. Source: WSO2

The communications between devices and cloud services or gateways involve different networking technologies.

Ethernet connects stationary or fixed IoT devices like security and video cameras, permanently installed industrial equipment, and gaming consoles.

WiFi, the most popular technology of wireless networking, is a great fit for data-intensive IoT solutions that are easy to recharge and operate within a small area. A good example of use is smart home devices connected to the electrical grid.

NFC (Near Field Communication) enables simple and safe data sharing between two devices over a distance of 4 inches (10 cm) or less.

Bluetooth is widely used by wearables for short-range communications. To meet the needs of low-power IoT devices, the Bluetooth Low-Energy (BLE) standard was designed. It transfers only small portions of data and doesn't work for large files.

LPWAN (Low-power Wide-area Network) was created specifically for IoT devices. It provides long-range wireless connectivity on low power consumption with a battery life of 10+ years. Sending data periodically in small portions, the technology meets the requirements of smart cities, smart buildings, and smart agriculture (field monitoring).

ZigBee is a low-power wireless network for carrying small data packages over short distances. The outstanding thing about ZigBee is that it can handle up to 65,000 nodes. Created specifically for home automation, it also works for low-power devices in industrial, scientific, and medical sites.

Cellular networks offer reliable data transfer and nearly global coverage. There are two cellular standards developed specifically for IoT things. LTE-M (Long Term Evolution for Machines) enables devices to communicate directly with the cloud and exchange high volumes of data. NB-IoT or Narrowband IoT uses low-frequency channels to send small data packages.

Major networking technologies used in the IoT projects.

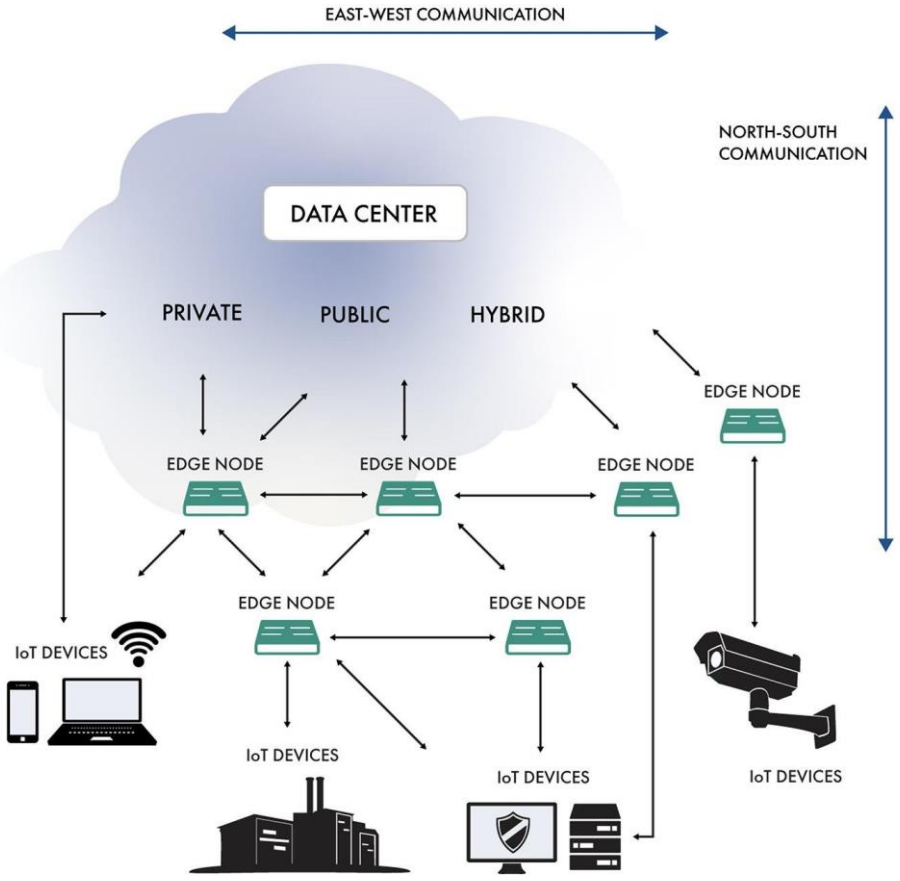
Once parts of the IoT solution are networked, they still need messaging protocols to share data across devices and with the cloud. The most popular protocols used in the IoT ecosystems are:

- **DDS (the Data Distribution Service)** which directly connects IoT things to each other and to applications addressing the requirements of real-time systems;
- **AMQP (the Advanced Message Queuing Protocol)** aiming at peer-to-peer data exchange between servers;
- **CoAP (the Constrained Application Protocol),** a software protocol designed for *constrained devices* — end nodes limited in memory and power (for example, wireless sensors). It feels much like HTTP but uses fewer resources;
- **MQTT (the Message Queue Telemetry Transport)**, a lightweight messaging protocol built on top of TCP/IP stack for centralized data collection from low-powered devices.

Edge or fog computing layer: reducing system latency

This level is essential for enabling IoT systems to meet the speed, security, and scale requirements of the 5th generation mobile network or 5G. The new wireless standard promises faster speeds, lower latency, and the ability to handle many more connected devices, than the current 4G standard.

The idea behind edge or fog computing is to process and store information as early and as close to its sources as possible. This approach allows for analyzing and transforming high volumes of real-time data locally, at the edge of the networks. Thus, you save the time and other resources that otherwise would be needed to send all data to cloud services. The result is reduced system latency that leads to real-time responses and enhanced performance.



The scheme of communications between IoT devices, edge nodes, and cloud data centers. Source: DesignNews

Edge computing occurs on gateways, local servers, or other edge nodes scattered across the network. At this level, data can be:

- evaluated to determine if it needs further processing at higher levels,
- formatted for further processing,
- decoded,
- filtered, and
- redirected to an additional destination

To sum up, the first three layers see data in motion, as it is constantly moving and altering. Only on hitting the next level, is data finally at rest and available for use by consumer applications.

Processing layer: making raw data useful

The processing layer accumulates, stores, and processes data that comes from the previous layer. All these tasks are commonly handled via IoT platforms and include two major stages.

Data accumulation stage

The real-time data is captured via an API and put at rest to meet the requirements of non-real-time applications. The data accumulation component stage works as a transit hub between event-based data generation and query-based data consumption.

Among other things, the stage defines whether data is relevant to the business requirements and where it should be placed. It saves data to a wide range of storage solutions, from data lakes capable of holding unstructured data like images and video streams to event stores and telemetry databases. The total goal is to sort out a large amount of diverse data and store it in the most efficient way.

Data abstraction stage

Here, data preparation is finalized so that consumer applications can use it to generate insights. The entire process involves the following steps:

- combining data from different sources, both IoT and non-IoT, including ERM, ERP, and CRM systems;
- reconciling multiple data formats; and
- aggregating data in one place or making it accessible regardless of location through data virtualization.

Similarly, data collected at the application layer is reformatted here for sending to the physical level so that devices can “understand” it.

Together, the data accumulation and abstraction stages veil details of the hardware, enhancing the interoperability of smart devices. What’s more, they let software developers focus on solving particular business tasks — rather than on delving into the specifications of devices from different vendors.

Application layer: addressing business requirements

At this layer, information is analyzed by software to give answers to key business questions. There are hundreds of IoT applications that vary in complexity and function, using different technology stacks and operating systems. Some examples are:

- device monitoring and control software,
- mobile apps for simple interactions,
- business intelligence services, and
- analytic solutions using machine learning.

Currently, applications can be built right on top of IoT platforms that offer software development infrastructure with ready-to-use instruments for data mining, advanced analytics, and data visualization. Otherwise, IoT applications use APIs to integrate with middleware.

Business layer: implementing data-driven solutions

The information generated at the previous layers brings value if only it results in problem-solving solution and achieving business goals. New data must initiate collaboration between stakeholders who in turn introduce new processes to enhance productivity.

The decision-making usually involves more than one person working with more than one software solution. For this reason, the business layer is defined as a separate stage, higher than a single application layer.

Security layer: preventing data breaches

It goes without saying that there should be a security layer covering all the above-mentioned layers. IoT security is a broad topic worthy of a separate article. Here we'll only point out the basic features of the safe architecture across different levels.

Device security. Modern manufacturers of IoT devices typically integrate security features both in the hardware and firmware installed on it. This includes

- embedded TPM (Trusted Platform Module) chips with cryptographic keys for authentication and protection of endpoint devices;
- a secure boot process that prevents unauthorized code from running on a powered-up device;
- updating security patches on a regular basis; and
- physical protection like metal shields to block physical access to the device.

Connection security. Whether data is being sent over devices, networks, or applications, it should be encrypted. Otherwise, sensitive information can be read by anybody who intercepts information in transit. IoT-centric messaging protocols like MQTT, AMQP, and DDS may use standard Transport Layer Security (TSL) cryptographic protocol to ensure end-to-end data protection.

Cloud security. Data at rest stored in the cloud must be encrypted as well to mitigate risks of exposing sensitive information to intruders. Cloud security also involves authentication and authorization mechanisms to limit access to the IoT applications. Another important security method is device identity management to verify the device's credibility before allowing it to connect to the cloud.

The good news is that IoT solutions from large providers like Microsoft, AWS, or Cisco come with pre-built protection measures including end-to-end data encryption, device authentication, and access control. However, it always pays to ensure that security is tight at all levels, from the tiniest devices to complex analytical systems.