**ASSIGNMENT 1**

|  |  |  |  |
| --- | --- | --- | --- |
| **Qualification** | **TEC Level 5 HND Diploma in Computing** | | |
| **Unit number and title** | **Unit 45: Internet of Things** | | |
| **Submission date** | 9/10/2023 | **Date Received 1st submission** |  |
| **Re-submission Date** |  | **Date Received 2nd submission** |  |
| **Student Name** | Tran Cong Hoang | **Student ID** | BH00317 |
| **Class** | IT0601 | **Assessor name** | Le Van Thuan |
| **Student declaration**  I certify that the assignment submission is entirely my own work and I fully understand the consequences of plagiarism. I understand that making a false declaration is a form of malpractice. | | | |
|  |  | **Student’s signature** | Conghoang.tran |

**Grading grid**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| P1 | P2 | M1 | M2 | D1 |
|  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| **❒ Summative Feedback: ❒ Resubmission Feedback:** | | |
| **Grade:** | **Assessor Signature:** | **Date:** |
| **Internal Verifier’s Comments:** | | |
| **Signature & Date:** | | |

|  |
| --- |
| Table of Contents  [**A.** **Introduction** 5](#_Toc148056895)  [**B.** **Contents** 6](#_Toc148056896)  [**P1: Explore various forms of IoT functionality.** 6](#_Toc148056897)  [**I.** **Introduction to IoT** 6](#_Toc148056898)  [1) Definition of IoT 6](#_Toc148056899)  [2) IoT standard 6](#_Toc148056900)  [3) Basic components of IoT 7](#_Toc148056901)  [**II.** **Basic functions of IoT** 8](#_Toc148056902)  [**III.** **History of birth and development of IoT.** 9](#_Toc148056903)  [1) Formative period (1990-2000) 9](#_Toc148056904)  [2) Development period (2000-2010) 10](#_Toc148056905)  [3) Boom period (2010-present) 11](#_Toc148056906)  [**IV.** **Characteristics and characteristics of IoT** 12](#_Toc148056907)  [1) Criteria used to evaluate IoT features and characteristics 12](#_Toc148056908)  [2) List and describe the features and characteristics of IoT according to each criterion 13](#_Toc148056909)  [3) Compare IoT with other technologies 13](#_Toc148056910)  [**V.** **Why is IoT widely used and of high importance in the current era?** 14](#_Toc148056911)  [**VI.** **Application of IOT** 15](#_Toc148056912)  [**P2: Review standard architecture, frameworks, tools, hardware, and APIs available for IoT development.** 20](#_Toc148056913)  [**I.** **Architecture of IoT** 20](#_Toc148056914)  [**1)** **Edge computing** 20](#_Toc148056915)  [**2)** **Cloud computing** 21](#_Toc148056916)  [**3)** **Hybrid computing** 22](#_Toc148056917)  [**4)** **4-layer model of IoT.** 23](#_Toc148056918)  [**5)** **Compare and evaluate the advantages and disadvantages of different architectures** 24](#_Toc148056919)  [**II.** **Framework of IoT** 26](#_Toc148056920)  [**1)** **What are IoT Frameworks? Some popular frameworks.** 26](#_Toc148056921)  [**2)** **Compare and evaluate the pros and cons of some popular frameworks.** 28](#_Toc148056922)  [**3)** **Real-life examples illustrate each frame.** 30](#_Toc148056923)  [**III.** **Standard Tools, How Arduino/Raspberry Pi boards and terminals connect and communicate.** 31](#_Toc148056924)  [**1)** **Tools to support IoT device programming and development.** 31](#_Toc148056925)  [**2)** **Arduino/Raspberry Pi board and end device.** 32](#_Toc148056926)  [**3)** **How to connect and communicate between Arduino and Raspberry Pi boards and terminals** 33](#_Toc148056927)  [**IV.** **Sensor.** 34](#_Toc148056928)  [**V.** **API standard for use in IOT development.** 35](#_Toc148056929)  [**C.** **Conclusion** 39](#_Toc148056930)  [**D.** **References** 41](#_Toc148056931) |

|  |
| --- |
| **List of figure**  [Figure 1: Internet of things 6](file:///C:\BTEC_FPT\IT0601-IoT\Assignments\TranCongHoang_BH00317_IoT_Assignment1_1st.docx#_Toc148056828)  [Figure 2: Components of IoT 7](file:///C:\BTEC_FPT\IT0601-IoT\Assignments\TranCongHoang_BH00317_IoT_Assignment1_1st.docx#_Toc148056829)  [Figure 3: Functions of IoT 8](file:///C:\BTEC_FPT\IT0601-IoT\Assignments\TranCongHoang_BH00317_IoT_Assignment1_1st.docx#_Toc148056830)  [Figure 4: Formative period (1990 -2000) 9](file:///C:\BTEC_FPT\IT0601-IoT\Assignments\TranCongHoang_BH00317_IoT_Assignment1_1st.docx#_Toc148056831)  [Figure 5: Development period of IoT(2000-2010) 10](file:///C:\BTEC_FPT\IT0601-IoT\Assignments\TranCongHoang_BH00317_IoT_Assignment1_1st.docx#_Toc148056832)  [Figure 6: Boom period of IoT 2010- prsent 11](file:///C:\BTEC_FPT\IT0601-IoT\Assignments\TranCongHoang_BH00317_IoT_Assignment1_1st.docx#_Toc148056833)  [Figure 7: Application of IoT 16](file:///C:\BTEC_FPT\IT0601-IoT\Assignments\TranCongHoang_BH00317_IoT_Assignment1_1st.docx#_Toc148056834)  [Figure 8: Smart home 17](file:///C:\BTEC_FPT\IT0601-IoT\Assignments\TranCongHoang_BH00317_IoT_Assignment1_1st.docx#_Toc148056835)  [Figure 9: Smart car 17](file:///C:\BTEC_FPT\IT0601-IoT\Assignments\TranCongHoang_BH00317_IoT_Assignment1_1st.docx#_Toc148056836)  [Figure 10: Smart health 18](file:///C:\BTEC_FPT\IT0601-IoT\Assignments\TranCongHoang_BH00317_IoT_Assignment1_1st.docx#_Toc148056837)  [Figure 11: Smart industry 18](file:///C:\BTEC_FPT\IT0601-IoT\Assignments\TranCongHoang_BH00317_IoT_Assignment1_1st.docx#_Toc148056838)  [Figure 12: Smart security 19](file:///C:\BTEC_FPT\IT0601-IoT\Assignments\TranCongHoang_BH00317_IoT_Assignment1_1st.docx#_Toc148056839)  [Figure 13: Framework of IoT 27](file:///C:\BTEC_FPT\IT0601-IoT\Assignments\TranCongHoang_BH00317_IoT_Assignment1_1st.docx#_Toc148056840)  [Figure 14: Arduino Uno 32](file:///C:\BTEC_FPT\IT0601-IoT\Assignments\TranCongHoang_BH00317_IoT_Assignment1_1st.docx#_Toc148056841) |

|  |
| --- |
| **List of table**  [Table 1: Criteria for evaluating IoT features and characteristics 14](#_Toc148056808)  [Table 2: Features and characteristics of IoT 14](#_Toc148056809)  [Table 3: Compare IoT with other technologies 15](#_Toc148056810)  [Table 4: Benefits, challenges and risks of IoT 16](#_Toc148056811)  [Table 5: Advantages and dis advantages of architectures 26](#_Toc148056812)  [Table 6: The pros and cons of IoT framework 31](#_Toc148056813) |

# **Introduction**

Internet of Things (IoT) is a technology that allows devices to connect and transmit data over the internet. IoT can be applied to many different fields such as health, education, agriculture, security, and defense. IoT is a broad concept, encompassing components such as architecture, frameworks, tools, hardware, and APIs. IoT architecture is the way IoT devices are organized and connected to cloud services. The IoT framework is a collection of technologies, standards, and protocols to support the development and management of IoT applications. IoT tools are pieces of software or hardware to create, test, and debug IoT applications. IoT hardware are physical or virtual devices capable of connecting to the internet and collecting, processing, and transmitting data. IoT APIs are application programming interfaces to enable IoT applications to interact with each other and with cloud services.

IoT is a new field and has been developing rapidly in recent times. However, IoT development also faces many challenges and issues, such as security, privacy, compatibility, performance, and cost. Much research has been conducted to address these issues, but there is still a lack of a general framework to compare and evaluate different forms of architectures, frameworks, tools, hardware, and APIs for development. Iodine. This report will attempt to fill this gap by analyzing and comparing these forms according to different criteria.

The research question of this paper is: Which forms of architecture, frameworks, tools, hardware, and APIs are best suited for different problem-solving requirements in IoT development? This question is based on the assumption that there is no single form that is optimal for all cases but depends on factors such as the goals, context, resources, and risks of each project. Iodine.

This report will use a comparative and evaluation approach to analyze different forms of architectures, frameworks, tools, hardware, and APIs for IoT development. This report will rely on criteria such as security, privacy, compatibility, performance, and cost to determine the pros and cons of these forms. This report seeks to provide recommendations for IoT developers when choosing appropriate formats for their projects.

# **Contents**

## **P1: Explore various forms of IoT functionality.**

### **Introduction to IoT**

#### Why the Internet of Things needs AIDefinition of IoT

Figure : Internet of things

IoT stands for Internet of Things, meaning Internet of Things. This is a system consisting of devices, machines, objects, animals, or people that are capable of connecting and transmitting data over the Internet without requiring human or human interaction. With computer. It is a way to describe IoT in scientific and precise language, based on standards and principles recognized by international or national organizations.

#### IoT standard

IoT standards: These are rules or guidelines established to ensure the compatibility, safety, and efficiency of IoT. IoT standards can relate to aspects such as architecture, protocols, security, privacy, quality, etc. IoT standards can be issued by international or national organizations, such as ISO/IEC JTC 1/SC 41, IEEE, ITU-T, ETSI, TCVN, etc.

Some examples of formal and standard definitions for IoT are:

- In 2013, the Global Standards Initiative on the Internet of Things (IoT-GSI) defined IoT as "a global infrastructure serving the information society, supporting intensive (computing) services through objects (both real and virtual) are interconnected using integrated existing information and communication technologies," and for that purpose, a "thing" is "a thing in the real world (a physical object) ) or information world (virtual object), which can be identified and integrated into a communication network".

- In 2015, the American National Standards Institute (NIST) defined IoT as "a network of devices, machines, people, and other entities connected to the internet through uniquely identifiable technologies, capable of collecting and exchanging data using available communication protocols".

- In 2019, the European Standards Agency (ETSI) defined IoT as "a global ecosystem for devices, machines and people capable of connecting, interacting and exchanging data".

#### Basic components of IoT

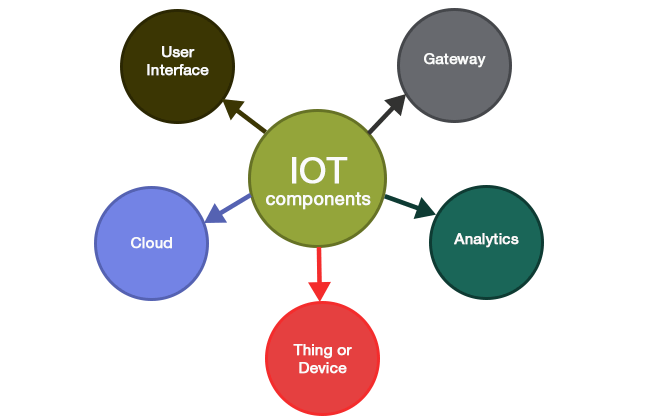
An IoT system usually includes the following components:

Figure : Components of IoT

* **Devices (or Things**): These are smart devices that can collect data from the environment using sensors or receive commands from users using controllers. For example: toothbrushes, vacuum cleaners, cars, and machinery.
* **Gateway**: These are devices that have the function of transmitting data from devices to the network infrastructure and vice versa. For example: routers, modems, and smartphones.
* **Network infrastructure (Network and Cloud)**: Network is the communication channels between devices and the cloud, such as Wi-Fi, Bluetooth, Zigbee, LoRaWAN, 5G,... Cloud is the services and platforms on The cloud allow to store, process, and manage data from devices. For example: AWS IoT, Google Cloud IoT, Microsoft Azure IoT.
* **Services-creation and Solution Layers**: These are applications and solutions that use data from network infrastructure to create business value or improve life. For example: consumption trend analysis, remote control, and warning system.
* **Applications**: Are software or services capable of providing solutions or functions for users or businesses based on data from IoT, such as smart home control applications, and health monitoring applications. health, smart production management applications, etc.

### **Basic functions of IoT**

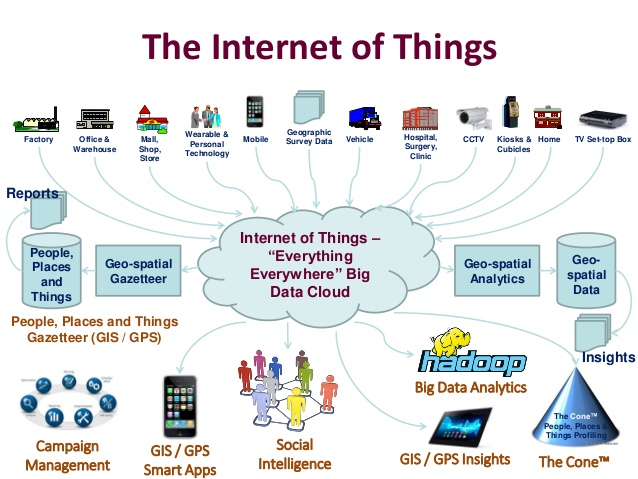
Basic IoT functions are the operations that IoT devices can perform with data, such as collection, processing, transmission, display, and control. These functions are performed by various components of IoT, including devices, networks, clouds, and applications. Here are some descriptions and examples of these functions and components:

Figure : Functions of IoT

* **Data collection**: This is the function of IoT devices to collect data from the surrounding environment or users. IoT devices use sensors or cameras to collect data about temperature, light, sound, images, and more. IoT devices can also receive data from users through interfaces such as keyboards, touchscreens, voice, etc. For example: A smartwatch can collect data about the user's heart rate, steps, and sleep.
* **Data processing**: This is the function of IoT devices or the cloud to process collected data. IoT devices can process data in a simple or complex way depending on their computing capabilities. IoT devices can use algorithms or artificial intelligence technology to process data. The cloud can handle larger and more complex data with the help of cloud computing platforms. For example, a security camera can process images to recognize faces or detect motion. The cloud can process images from multiple cameras to analyze people's behavior or trends.
* **Data transfer**: This is the function of IoT devices and networks to transfer data from device to cloud or from cloud to device. IoT devices and networks use different communication protocols and technologies to transmit data over the internet. Communication protocols and technologies can be wired or wireless, for example, Wi-Fi, Bluetooth, Zigbee, LoRaWAN, 5G, etc. For example, a temperature sensor can transmit temperature data over Wi-Fi to the cloud. The cloud can transmit control commands via 5G to an air conditioner to adjust the temperature.
* **Data display**: This is the function of IoT devices or applications to display data to users. IoT devices can display data on screens such as LED, LCD, OLED, etc. Applications can display data on mobile devices or computers through graphical user interfaces. For example: A smart watch can display data about the user's heart rate, steps, and sleep on an OLED screen. A mobile app can display smart home temperature, humidity, and air quality data on a touch screen.
* **Device control**: This is the function of IoT devices or applications to control other devices. IoT devices can control other devices through interfaces such as switches, buttons, voice, etc. Applications can control other devices through commands or settings on mobile devices or computers. For example: A smart speaker can control other smart home devices by voice. A mobile app can control a smart car to start, unlock, or find its location.

### **History of birth and development of IoT.**

The history of the birth and development of IoT can be divided into three main periods: the formation period (1990-2000), the development period (2000-2010) and the boom period (2010-2023). Here are some important events, characters, and technologies from each period:

#### Formative period (1990-2000)

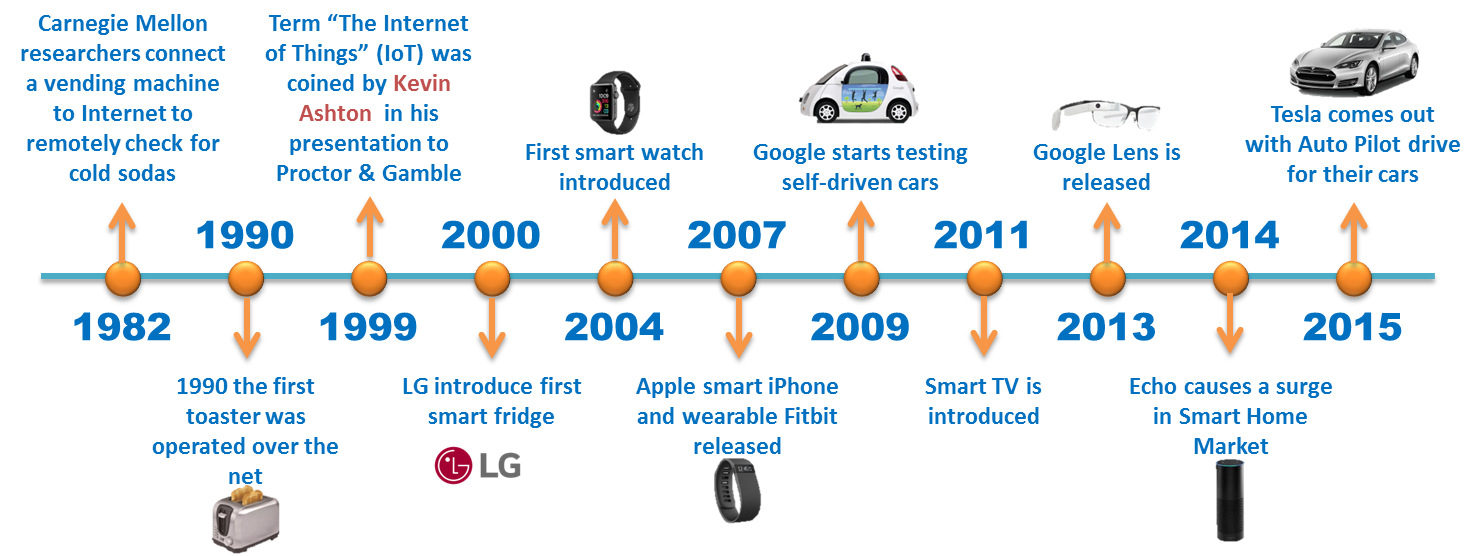
This is the stage where the first IoT devices are created and connected to the internet, mainly for research or entertainment purposes. Some notable events, characters, and technologies from this period are:

Figure : Formative period (1990 -2000)

* In 1989, David Nichols and colleagues at MIT invented the first IoT device, a Coca-Cola vending machine that could check the quantity and temperature of soft drink cans over the internet.
* In 1990, John Romkey and Simon Hackett created the first internet-connected toast, which could be turned on and off remotely.
* In 1993, Quentin Stafford-Fraser and colleagues at Cambridge University installed a camera near the coffee pot to monitor the remaining amount of coffee via the Internet. This is the world's first online webcam.
* In 1995, Steve Mann invented WearCam, the first wearable IoT device, a precursor to today's wearable technology devices.
* In 1999, Kevin Ashton, director of Auto-ID Labs at MIT, used the term "Internet of Things" in a presentation to Procter & Gamble executives to illustrate the potential of RFID technology.

#### Development period (2000-2010)

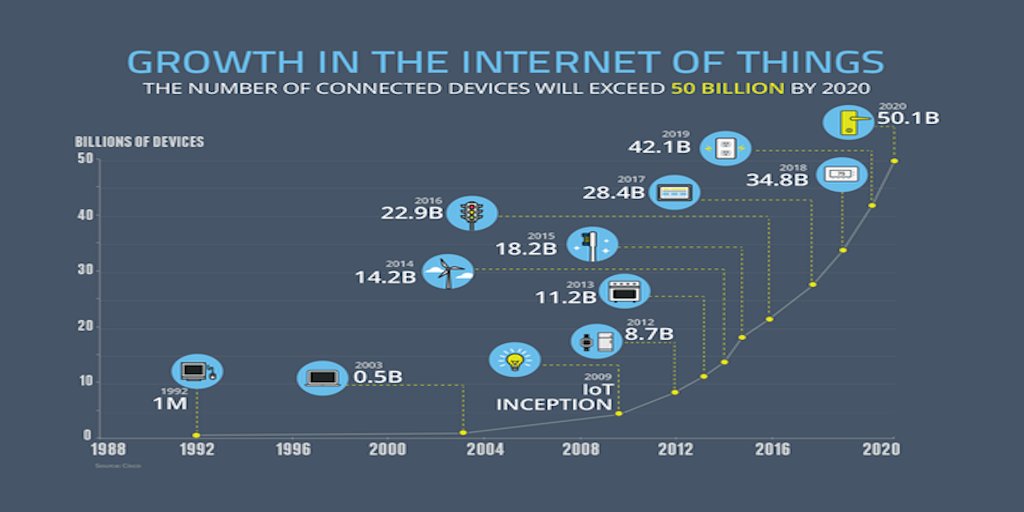
This is the period when IoT devices are improved and more popular, thanks to the development of technologies such as cloud computing, IPv6, GPS, RFID, Zigbee, etc. Some notable events, characters, and technologies from this period are:

Figure : Development period of IoT(2000-2010)

* In 2000, LG introduced the first smart refrigerator that could connect to the internet and could manage food, send emails, and view news.
* In 2002, Microsoft launched Windows CE .NET, an operating system for embedded and IoT devices.
* In 2005, Cisco Systems initiated the Planetary Skin project to collect data from millions of sensors globally to monitor climate change.
* In 2008, IBM launched the Smarter Planet program to encourage the use of IoT to solve social and economic problems.
* In 2009, Google launched the PowerMeter service, allowing users to monitor and control the electricity consumption of home devices via the internet.

#### Boom period (2010-present)

This is the period when IoT devices become very popular and diverse, thanks to the decline in prices of components, the acceleration of wireless connections, the support of cloud computing platforms, and the interest of both consumers and businesses. Some notable events, characters, and technologies from this period are:

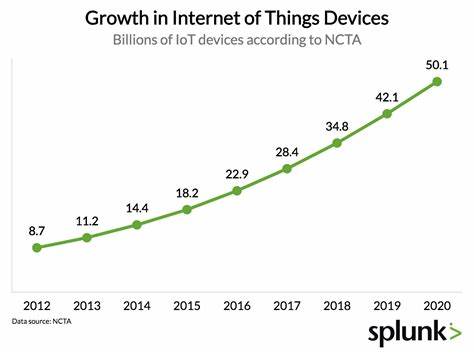
* In 2011, Nest Labs launched the Nest Learning Thermostat, an IoT device capable of self-adjusting temperature based on user behavior and can be controlled remotely via the internet.

Figure : Boom period of IoT 2010- prsent

* In 2013, Google launched Google Glass, a smart glass that can connect to the internet and display information in the user's eyes.
* In 2014, Amazon launched Echo, a smart speaker that can connect to the internet and has voice control capabilities through the virtual assistant Alexa.
* In 2015, Samsung launched the Samsung Gear S2, a smartwatch that can connect to the internet and can monitor health, receive notifications and make payments.
* In 2016, Tesla launched the Tesla Model S, a smart car that can connect to the internet and is capable of self-driving, self-parking, and software updates.
* In 2017, Apple launched the Apple Watch Series 3, a smartwatch that can connect to the internet without a smartphone and is capable of making phone calls, listening to music, and using Siri.
* In 2018, Amazon launched Ring Door View Cam, a smart security camera that can connect to the internet and allow users to see who is knocking on the door or making deliveries via smartphone or Echo speaker.
* In 2019, Google launched Stadia, an online gaming service over the internet that allows users to play games on any device that can run Chrome or YouTube.
* In 2020, Samsung launched the Galaxy Z Flip, a foldable smartphone with internet connectivity and advanced features such as dual cameras, in-display fingerprint, and Bixby virtual assistant.
* In 2021, Facebook launched Ray-Ban Stories, smart glasses in partnership with Ray-Ban that can connect to the internet and allow users to take photos, record videos, listen to music, and make phone calls via Facebook Assistant.
* 2022: Microsoft launches HoloLens 3, a smart VR/AR device that can connect to the internet and allows users to experience vivid images and sounds in three-dimensional space.

### **Characteristics and characteristics of IoT**

#### Criteria used to evaluate IoT features and characteristics

The criteria used to evaluate IoT characteristics and characteristics are important factors that influence the quality, efficiency, and applicability of IoT devices in different contexts. These criteria include:

* **Size**: Bigness, smallness, thickness, thinness, weight, capacity, surface area, shape, structure of the IoT device.
* **Connectivity**: Network type, frequency, bandwidth, latency, security, reliability, coverage of IoT devices when communicating with other devices or the internet.
* **Interoperability**: Protocol type, standard type, interface type, user type of IoT devices when operating with other devices or with users according to common protocols and standards.
* **Self-management ability**: Type of problem, type of solution, type of environment of IoT devices when self-adjusting or self-repairing when encountering problems or when the environment changes.
* **Learning ability**: The type of data, the type of algorithm, the type of goal of an IoT device when improving or developing itself when there is new data or when there is feedback from the user or from the environment.

The criteria used to evaluate IoT characteristics and characteristics are important factors that influence the quality, efficiency, and applicability of IoT devices in different contexts.

The purpose of using these criteria is to be able to compare, classify and select IoT devices that suit specific needs and requirements.

Specific factors related to each criterion can be listed as follows:

|  |  |
| --- | --- |
| **Criteria** | **Specific factors** |
| Size | Largeness, smallness, thickness, thinness, weight, capacity, surface area, shape, texture |
| Connectivity | Network type (e.g. Wi-Fi, Bluetooth, ZigBee, LoRaWAN), frequency, bandwidth, latency, security, reliability, coverage |
| Interoperability | Protocol type (e.g. MQTT, CoAP, HTTP), standard type (e.g. IEEE 802.15.4, OCF, oneM2M), interface type (e.g. LED, LCD, audio, vibration), user type use (e.g. people, machines) |
| Self-management ability | Problem type (e.g. hardware error, software error, network error), solution type (e.g. auto-restart, auto-update, self-reporting), environment type (e.g. temperature, light, humidity) |
| Ability to learn | Type of data (e.g. figures, text, images), type of algorithm (e.g. supervised learning, unsupervised learning, reinforcement learning), type of goal (e.g. classification, retrieval rule, group) |

Table 1: Criteria for evaluating IoT features and characteristics

#### List and describe the features and characteristics of IoT according to each criterion

|  |  |  |
| --- | --- | --- |
| **Criteria** | **Characteristic** | **Featured** |
| Size | The size of an IoT device can range from very small (e.g. sensors) to very large (e.g. self-driving cars) | The size of an IoT device affects mobility, savings energy and flexibility |
| Connectivity | Connectivity is the ability of an IoT device to communicate with other devices or the internet | IoT device connectivity affects performance, safety, and continuity |
| Interoperability | Interoperability is the ability of an IoT device to operate with other devices or with users according to common protocols and standards | IoT device interoperability affects compatibility, extensibility, and innovation |
| Self-management capability | Self-management is the ability of an IoT device to self-regulate or repair itself when problems occur or when the environment changes | Self-management of IoT devices affects stability, flexibility and sustainability |
| Learnability | Learning capability is the ability of an IoT device to self-improve or evolve as new data becomes available or as feedback is received from the user or from the environment | The learning ability of IoT devices affects intelligence, personalization and innovation |

Table 2: Features and characteristics of IoT

#### Compare IoT with other technologies

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Technology** | **Definition** | **Advantages** | **Disadvantages** | **Comparison with IoT** |
| Traditional Internet | The traditional Internet is a global network connecting computers and devices through TCP/IP protocols | The traditional Internet allows access and sharing of data, information and services worldwide | Traditional Internet can encounter problems with security, privacy and quality of service | IoT is part of the traditional internet, but is not limited to connecting computers and devices, but also connects objects in everyday life |
| Wireless Network | A wireless network is a network that connects devices via radio waves, without the need for cables or fixed systems | Wireless networks allow for more flexible, mobile and convenient connections than wired networks | Wireless networks may have more speed, reliability and security issues than wired networks | IoT uses different types of wireless networks to connect IoT devices, depending on distance, capacity and power consumption requirements |
| Cloud Computing | Cloud computing is a model that provides computer resources (such as CPU, RAM, hard drive, software) on demand via the internet | Cloud computing allows for cost savings, increased performance and scalability compared to traditional computing | Cloud computing may face more security, privacy, and service provider dependency issues than traditional computing | IoT uses cloud computing to store, process, and analyze data collected from IoT devices, as well as to remotely manage and control IoT devices |

Table 3: Compare IoT with other technologies

### **Why is IoT widely used and of high importance in the current era?**

|  |  |  |
| --- | --- | --- |
| **Object** | **Benefits** | **Challenges and risks |** |
| Personal | - Improve quality of life through smart devices such as smart watches, smart homes, smart cars.  - Save time and costs through automating daily activities such as laundry, cooking, and shopping.  - Improve health and safety through monitoring body indicators, early detection of diseases, and emergency warnings. | - Loss of privacy and security of personal information due to data disclosure or theft from IoT devices.  - Loss of control and autonomy due to overreliance on IoT devices.  - Loss of skills and creativity due to passivity of daily activities. |
| Organization | - Increase performance and revenue through collecting and analyzing data from IoT devices to improve products and services and find new revenue sources  - Reduce costs and risks through remote monitoring and control of IoT devices to prevent or fix problems and save energy.  - Improve quality and reputation through providing customized IoT solutions according to customer needs and desires | - Loss of competition and profits due to the inability to meet customer needs and expectations for IoT solutions.  - Loss of reputation and quality due to encountering technical, security or legal problems related to IoT devices.  - Loss of manpower and resources due to insufficient skills, knowledge and investment to deploy and maintain IoT solutions. |
| Society | - Improve the environment and sustainability through reducing emissions, waste and resource consumption using IoT.  - Promote development and social justice through connecting and supporting communities with little access to technology, education, and healthcare.  - Enhance national security and safety through the use of IoT to prevent crime, terrorism, and natural disasters. | - Social imbalance and injustice due to the lack of equal and fair distribution of opportunities and benefits from IoT.  - Loss of national safety and security due to attacks, intrusions or taking advantage of IoT devices to harm people, property or power.  - Loss of the environment and sustainability due to the generation of too much e-waste, too much energy consumption or pollution from IoT devices. |

Table 4: Benefits, challenges and risks of IoT

### **Application of IOT**

1. Definition of IoT application domains as specific areas or goals where IoT can provide smart, efficient, and innovative solutions.

* IoT application domains are the industries or social causes that IoT can contribute to by connecting devices, data, and applications through the internet.
* Application domains of IoT can solve user problems, needs, and desires using advanced technologies such as sensors, cloud computing, artificial intelligence, machine learning, and blockchain,…
* IoT application domains can create new products and services or improve existing products and services by enhancing the connectivity, interaction, self-management, and learning capabilities of devices. bag.

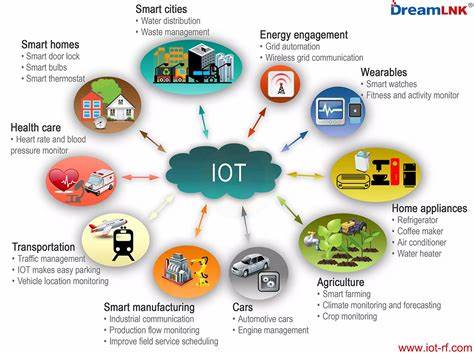
1. Explain the importance of the application domains of IoT that they can bring many benefits to individuals, organizations and society, such as improving quality of life, increasing business performance, minimizing impacts environment.

Figure : Application of IoT

* Application domains of IoT can bring many benefits to individuals, for example:
* Improves quality of life by helping users save time, costs and energy in daily activities such as controlling home devices, monitoring health, navigating traffic,...
* Enhance comfort and safety by helping users control and automate devices as desired, detect and warn of incidents or dangers,...
* Enhance creativity and learning by helping users access and share information, knowledge and skills from different sources,...
* Application domains of IoT can bring many benefits to organizations, for example:
* Increase performance and revenue by helping organizations collect and analyze data from IoT devices to improve products and services, find new revenue sources, etc.
* Reduce costs and risks by helping organizations remotely monitor and control IoT devices to prevent or fix problems, save energy, etc.
* Enhance quality and reputation by helping organizations provide IoT solutions customized to customer needs and desires,…
* Application domains of IoT can bring many benefits to society, for example:
  + Improve the environment and sustainability by helping society reduce emissions, waste and resource consumption due to the use of IoT,...
  + Promote development and social justice by helping society connect and support communities with little access to technology, education, healthcare, etc.
  + Enhance national security and safety by helping society use IoT to prevent crime, terrorism, natural disasters,...

1. Application areas of IoT according to specific industries or goals

* **Smart home**: Smart home is an application domain of IoT that allows users to control and automate home devices, such as lights, air conditioners, washing machines, security cameras, etc. An example of a smart home Smart is Google Nest, a smart home platform that allows users to connect and control home devices via voice or phone5. Smart homes can bring many benefits to users, such as saving time, costs and energy, improving comfort and safety, and enhancing creativity and learning.

Figure : Smart home

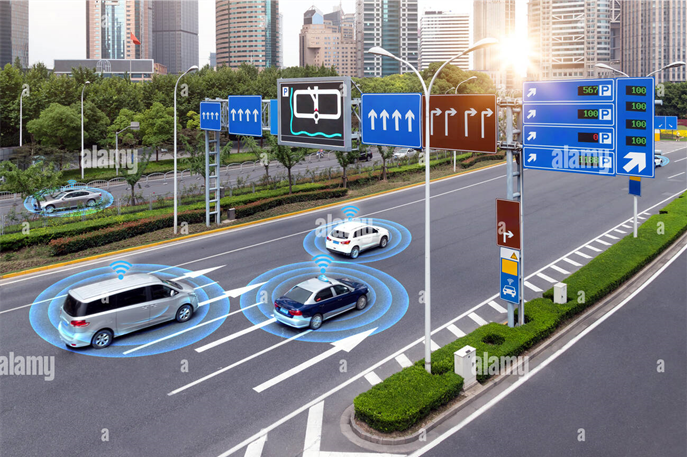
* **Smart vehicles**: Smart vehicles are an application domain of IoT that allows vehicles to connect to the Internet, communicate with other vehicles or traffic infrastructure, and self-driving or self-adjusting. An example of a smart car is the Tesla Model S, a smart electric car that allows users to access the Internet, stream music, use the global positioning system (GPS), and enable self-driving mode. Smart cars can bring many benefits to users, such as improving travel quality, reducing traffic accidents, and protecting the environment.

Figure : Smart car

* **Smart health**: Smart health is an application domain of IoT that allows users to monitor and improve their health, such as measuring heart rate, blood pressure, blood sugar, blood oxygen level. An example of smart health is Fitbit, a smart wearable device that allows users to track physical activity, sleep, calories consumed and burned. Smart healthcare can bring many benefits to users, such as:
* Improve health and prevent disease by helping users know their health status, receive advice and warnings from medical experts or smart algorithms.
* Save costs and time by helping users access medical services remotely, without having to go to a hospital or clinic unless absolutely necessary.
* Boost confidence and happiness by helping users set and track personal health goals, participate in health-related communities and competitions.

Figure : Smart health

* **Smart industry**: Smart industry is an application domain of IoT that allows businesses to improve production efficiency and quality, such as monitoring and controlling machines, production lines, and warehouses. An example of smart industry is Siemens MindSphere, an industrial IoT platform that allows businesses to connect and manage their devices, data and applications in the cloud. Smart industry can bring many benefits to businesses, such as:
* Increase performance and revenue by helping businesses collect and analyze data from IoT devices to improve products and services and find new revenue sources.

Figure : Smart industry

* Reduce costs and risks by helping businesses remotely monitor and control IoT devices to prevent or fix problems and save energy.
* Improve quality and reputation by helping businesses provide IoT solutions customized to customers' needs and desires.
* **Smart security**: Smart security is an application domain of IoT that allows users to protect themselves, their families and assets, such as intrusion detection, fire warning, and rescue calls. An example of smart security is Ring, a smart security system that allows users to see and talk to people coming to their door via a video doorbell connected to their phone. Smart security can bring many benefits to users, such as:

Figure : Smart security

* Enhances safety and peace of mind by helping users detect and promptly handle dangerous or emergency situations.
* Save costs and time by helping users monitor and control security devices remotely, without the need to install or maintain cables or batteries.
* Increase convenience and flexibility by helping users connect and interact with security devices via voice, phone or app.

## **P2: Review standard architecture, frameworks, tools, hardware, and APIs available for IoT development.**

### **Architecture of IoT**

#### **Edge computing**

##### What is edge computing and why is it important?

Edge computing is a distributed computing architecture in which data processing is performed close to where the data is created, instead of sent to the cloud or a central server. Edge computing can improve response time and save bandwidth for IoT applications. Edge computing can also increase security and privacy for users.

How edge computing works through a simple example. An example of edge computing is a smart security camera that can detect and warn of unusual activities without an internet connection. The camera can use an embedded processor to analyze images and sounds from the sensor, and send notifications to the user via a mobile app. This camera does not need to send all raw data to the cloud, only sending important or processed data.

##### The benefits and challenges of edge computing.

* The benefits of edge computing include:
* **Performance**: Edge computing can process data faster and more accurately by reducing the distance and number of intermediate steps between devices and the cloud. For example, edge computing can help save energy by adjusting lighting and temperature according to the user's schedule and preferences.
* **Latency**: Edge computing can reduce latency by sending and receiving data in real time, without having to wait for the cloud to respond¹. This is important for IoT applications that require instantaneous interaction, like self-driving cars or industrial robots.
* **Bandwidth**: Edge computing can save bandwidth by sending only necessary or processed data to the cloud, without sending the entire raw data. This can reduce costs and increase scalability for IoT applications.
* **Security**: Edge computing can increase security by reducing the risk of data being lost or corrupted when transmitted over the network. This can also help protect users' privacy by giving them control over their data.
* The challenges of edge computing include:
* **Management**: Edge computing can be more difficult to manage and integrate than cloud computing, due to the large number and diversity of devices and data at the network edge. This requires tools and frameworks to monitor, update, and control devices remotely.
* **Consistency**: Edge computing can have difficulty ensuring data consistency and synchronization across devices and the cloud. This can affect the quality and reliability of IoT applications.
* **Resources**: Edge computing can be limited by the resources of devices at the edge of the network, such as storage capacity, memory, battery and processing power. This may require load balancing and optimization solutions to distribute work effectively.

#### **Cloud computing**

##### What is cloud computing and why is it important?

Cloud computing is the provision of computing, storage and networking resources over the internet, without the need for users to manage and maintain physical devices themselves¹. Cloud computing can help save costs, increase efficiency, scalability and updates for IoT applications.

A simple example of how cloud computing works. A smart home system can control home appliances remotely via a mobile application. Users can connect devices such as lights, air conditioners, washing machines to the internet via Wi-Fi or Bluetooth. These devices send data about their status and activity to cloud servers over the internet. Cloud servers will store, process and analyze data to provide services such as device management, power optimization, security and notifications. Users can access and use these services through a mobile application, without having to install or configure anything on their device.

##### Benefits and challenges of cloud computing.

The benefits of cloud computing include:

* **Scalability**: Cloud computing can provide unlimited and flexible resources for IoT development. Cloud computing can react and adapt to changing needs by automatically allocating and reclaiming resources. This can reduce costs and increase the overall efficiency of IoT applications. For example, cloud computing can help an online sales website withstand a sudden increase in customers during the holidays.
* **Update**: Cloud computing can provide the latest hardware, software and services for IoT development. Cloud computing can update and upgrade resources automatically and continuously, without user intervention. For example, cloud computing can help a health monitoring system use the latest artificial intelligence technologies to detect and diagnose diseases.
* **Simplification**: Cloud computing can simplify the process of managing and deploying IoT applications. Cloud computing allows users to access and use resources through a web or mobile interface, without the need for installation or configuration. Cloud computing also allows users to focus on business logic, not on technical details.

Cloud computing challenges include:

* **Latency**: Cloud computing can cause delays in sending and receiving data between devices and the cloud. This can impact the performance and user experience of IoT applications that require instant interaction, like games or videos.
* **Security**: Cloud computing can leak or lose data when transmitted over the internet. This can pose security and privacy risks to users. For example, a smart camera can be hacked to spy on or secretly record users.

#### **Hybrid computing**

##### What is hybrid computing and why is it important?

Hybrid computing is a type of computing that combines both digital and analog components. Digital components are often used to perform logic and arithmetic operations, while analog components are often used to solve complex mathematical problems. Hybrid computing can bring advantages such as high speed, accuracy and flexibility to different applications.

A simple example of how hybrid computing works. A computer controls a nuclear reactor. The digital component of the computer monitors and controls the reactor's parameters, while the analog component of the computer simulates the reactor's behavior and predicts its future state.

##### The benefits and challenges of hybrid computing.

The benefits of hybrid computing include:

* **Speed**: Hybrid computing can solve mathematical problems faster than digital computers, by using analog components capable of parallel and continuous processing. For example, hybrid computing can help process medical images by using analog components to reconstruct, filter, and analyze images.
* **Accuracy**: Hybrid computing can solve mathematical problems more accurately than analog computers, by using digital components capable of representing and examining discrete values. For example, hybrid computing can help simulate weather systems using a numerical component to handle differential equations and numerical methods to model atmospheric dynamics.
* **Flexible**: Hybrid computing can be customized according to the needs of each application, by combining digital and analog components in different ratios. For example, hybrid computing can help optimize performance and cost for IoT applications by using digital components to manage and control devices, while using analog components to process data. from sensors.

The challenges of hybrid computing include:

* **Complexity**: Hybrid computing can be more difficult to design, deploy and maintain than individual digital or analog computers, due to differences in characteristics, communication and synchronization between components. This requires in-depth knowledge and skills of both types of computers.
* **Compatibility**: Hybrid computing can have difficulty in compatibility and integration with other systems, due to the lack of standardization and diversity of digital and analog components. This may require conversion protocols and interfaces to convert between digital and analog forms of data.

#### **4-layer model of IoT.**

##### What is the 4-tier model and why is it used?

The 4-tier model is a model that stratifies IoT according to the function of each component. This model is used to help better understand the process of data collection, transmission, processing, and application in IoT.

The layers of the 4-tier model are:

- Device layer: A layer consisting of devices, sensors and controllers capable of connecting to the internet and collecting data from the surrounding environment. This layer can also perform some basic processing and control of devices at the network edge.

- Network layer: This layer includes means of transmitting data between devices and the cloud. This layer can use wireless or wired protocols, such as Wi-Fi, Bluetooth, Zigbee, LoRaWAN, Ethernet, or USB. This layer can also use gateways to aggregate and filter data from multiple devices.

- Processing layer: A layer consisting of servers and cloud platforms capable of storing, processing and analyzing data from devices. This layer can use technologies such as machine learning, artificial intelligence, big data and blockchain to create value from data.

- Application layer: This layer includes software and interfaces that allow users to interact with the IoT system. This layer can run on mobile devices, personal computers or web browsers. This layer can display, control, and customize IoT devices and data.

##### Compare and evaluate the advantages and disadvantages of the 4-tier model compared to other architectures.

Some advantages and disadvantages of the 4-tier model are:

- Advantages: The 4-tier model can help simplify the process of designing, deploying and managing IoT applications. This model can also help clearly differentiate the roles and responsibilities of each component in an IoT system. This model can also help optimize the performance, security, and scalability of IoT systems.

- Disadvantages: The 4-tier model can have difficulty integrating and linking different components in the IoT system. This model may also not be suitable for some IoT applications that require more flexibility and architectural diversity. This model may also not meet some of the latency, bandwidth, or energy efficiency requirements of some IoT applications.

##### Real-life examples for the 4-tier model.

Some practical examples for the 4-tier model are:

- A system to monitor air quality in the city. The devices are air sensors mounted on public buses or bicycles. The sensors collect data on the concentration of pollutants in the air and send it to the gateway via Bluetooth. Gateways are Wi-Fi access points installed on bus stops or public spots. Gateways aggregate and filter data from multiple sensors and send it to the cloud over the internet. The cloud is where data is stored, processed and analyzed to create air quality reports, charts and alerts. Applications are software that allow users to view, share and receive notifications about air quality via phone or computer.

- A smart warehouse management system. The devices are barcode, RFID or NFC scanners mounted on products, pallets or containers. The devices collect data about the location, quantity and status of the goods and send them to the gateway via Wi-Fi or Zigbee. Gateways are embedded computers or personal computers connected to the warehouse's LAN. Gateways aggregate and process data from multiple devices and send it to the cloud over the internet. The cloud is where data is stored, processed and analyzed to create reports, charts and alerts about the warehouse situation. Applications are software that allow users to view, control and customize goods via phone or computer.

#### **Compare and evaluate the advantages and disadvantages of different architectures**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Criteria** | **Edge computing** | **Cloud computing** | **Hybrid computing** | **4-storey model** |
| Performance | Highest, due to reduced distance and number of intermediate steps between devices and the cloud. | Lowest, due to having to transmit data over the internet and wait for the cloud to respond. | Second highest, due to combining digital and analog components to take advantage of both. | Third highest, due to clearly separating the functions of each floor and optimizing the work of each floor. |
| Latency | Lowest, due to sending and receiving data in real time, there is no need to wait for the cloud. | Highest, due to having to transmit data over the internet and wait for the cloud. | Second lowest, due to the use of analog components capable of continuous processing. | Third lowest, due to the use of wireless or wired protocols to transmit data quickly. |
| Bandwidth | Lowest, since only necessary or processed data is sent to the cloud, there is no need to send the entire raw data. | Highest, due to having to send all raw data to the cloud. | Second lowest, due to the use of numeric components to represent and test discrete values. | Third lowest, due to use of gateways to aggregate and filter data from multiple devices. |
| Security | Highest, due to reducing the risk of data being lost or damaged when transmitted over the network. | Lowest, due to having to transmit data over the internet and depending on cloud service providers. | Second highest, due to the use of analog components capable of encoding and decoding data. | Third highest, due to use of security protocols for safe data transmission. |
| Scalability | Lowest, due to being limited by the resources of devices at the edge of the network. | Highest, due to providing unlimited and flexible resources for IoT development. | Second highest, due to customization according to the needs of each IoT application. | Third highest, due to the use of cloud platforms capable of automatically allocating and reclaiming resources. |
| Update | Lowest, due to difficulty in managing and updating devices at the network edge. | Highest, due to providing the latest resources and automatically updating and upgrading resources. | Second highest, due to combining digital and analog components to use the latest technologies. | Third highest, due to use of tools and frameworks to monitor, update and control remote devices. |
| Simplify | Second highest, due to allowing users to focus on business logic, not on technical details. | Best of all, because it allows users to access and use resources through a web or mobile interface, with no installation or configuration required. | Lowest, due to having to design, implement, and maintain individual digital and analog components. | Second lowest, due to having to differentiate and integrate different layers in the IoT system. |

Table : Advantages and dis advantages of architectures

Below are real-life examples of four popular IoT architectures:

- **Edge computing**: An example of edge computing is a smart security camera that can detect and warn of unusual activities without an internet connection. The camera can use an embedded processor to analyze images and sounds from the sensor, and send notifications to the user via a mobile app. This camera does not need to send all raw data to the cloud, only sending important or processed data.

- **Cloud computing**: An example of cloud computing is a smart home system that can control home appliances remotely via a mobile application. Users can connect devices such as lights, air conditioners, washing machines to the internet via Wi-Fi or Bluetooth. These devices send data about their status and activity to cloud servers over the internet. Cloud servers will store, process and analyze data to provide services such as device management, power optimization, security and notifications. Users can access and use these services through a mobile application, without having to install or configure anything on their device.

- **Hybrid computing**: An example of hybrid computing is a health monitoring system that can use edge computing to process biological data from smart wearable devices, and use cloud computing to store and analyze and share data with doctors or relatives. Users can connect smart wearable devices to phones or computers via Bluetooth. These devices will collect

data about the user's heart rate, blood pressure, body temperature and other indicators. These devices will use an analog chip to process biological data and generate digital values. These numeric values will be sent to your phone or computer via Bluetooth. The phone or computer will use a mobile application to display, monitor and control smart wearable devices. This mobile application will also send processed data to cloud servers over the internet. Cloud servers will store, analyze and share data with doctors or the user's relatives via email or text message. Doctors or relatives can view, receive notifications and advise users via a web or mobile application.

- **4-tier model**: An example of a 4-tier model is a system that monitors air quality in a city. The devices are air sensors mounted on public buses or bicycles. The sensors collect data on the concentration of pollutants in the air and send it to the gateway via Bluetooth. Gateways are Wi-Fi access points installed on bus stops or public spots. Gateways aggregate and filter data from multiple sensors and send it to the cloud over the internet. The cloud is where data is stored, processed and analyzed to create air quality reports, charts and alerts. Applications are software that allow users to view, share and receive notifications about air quality via phone or computer.

### **Framework of IoT**

#### **What are IoT Frameworks? Some popular frameworks.**

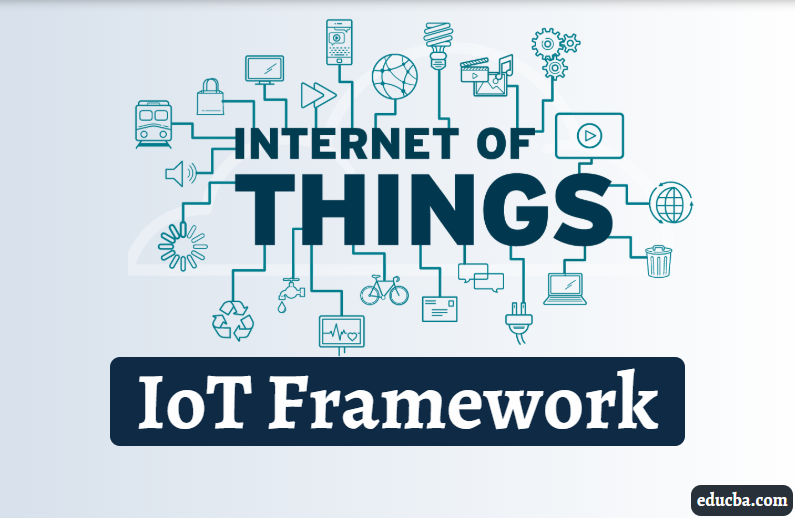
An IoT framework is a set of software tools and services that enable developers and businesses to build, manage, and optimize IoT solutions. An IoT framework usually includes the following components:

Figure : Framework of IoT

* A connectivity platform to enable IoT devices to communicate with each other and the cloud over standard protocols such as MQTT, HTTP or CoAP.
* A management platform to enable remote registration, organization, monitoring and control of IoT devices at scale.
* An analytics platform to enable collection, storage, processing and extraction of value from real-time or historical IoT data.
* An application platform to enable the creation, deployment, and operation of IoT applications in the cloud or edge, using technologies such as AI, ML, blockchain, or AR/VR.

Some typical IoT frameworks are:

- AWS IoT: Is the IoT framework of Amazon Web Services, providing services and solutions to connect and manage billions of IoT devices. AWS IoT includes components such as AWS IoT Core, AWS IoT Device Management, AWS IoT Analytics, AWS IoT Greengrass, and AWS IoT Device Defender.

- Azure IoT: Is Microsoft Azure's IoT framework, providing services and solutions to build IoT applications from edge to cloud. Azure IoT includes components such as Azure IoT Hub, Azure IoT Central, Azure IoT Edge, Azure Stream Analytics, and Azure Digital Twins.

- Google Cloud IoT: Is the IoT framework of Google Cloud Platform, providing services and solutions to connect, monitor and control billions of IoT devices. Google Cloud IoT includes components such as Google Cloud Pub/Sub, Google Cloud Functions, Google Cloud Dataflow, Google Cloud BigQuery, and Google Cloud ML Engine.

- IBM Watson IoT: Is IBM Cloud's IoT framework, providing services and solutions to collect and analyze data from billions of smart devices. IBM Watson IoT includes components such as IBM Watson IoT Platform, IBM Watson Studio, IBM Watson Assistant, IBM Maximo Asset Management and IBM Blockchain.

#### **Compare and evaluate the pros and cons of some popular frameworks.**

IoT frameworks can be compared and evaluated according to the following criteria:

* **Connectivity**: Is the ability to support a variety of devices and protocols to connect to the cloud or network edge.
* **Manageability**: Is the ability to provide tools and features to manage the status, configuration, security and updates of IoT devices.
* **Analytical ability**: Is the ability to process and query data from IoT devices to create useful information for decision making and action.
* **Applicability**: Is the ability to create, deploy and operate IoT applications in the cloud or network edge, using advanced technologies such as AI, ML, blockchain or AR/VR.
* **Integration ability**: Is the ability to connect and interact with other services and solutions of the same provider or of a third party.

The following table compares and evaluates the pros and cons of IoT frameworks according to the above criteria:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Criteria | AWS IoT | Azure IoT | Google Cloud IoT | IBM Watson IoT |
| Connectivity | High, supports MQTT, HTTP, WebSocket, LoRaWAN and custom protocols. | High, supports MQTT, AMQP, HTTP and custom protocols. | High, supports MQTT, HTTP and custom protocols. | High, supports MQTT, HTTP and custom protocols. |
| Management ability | Cao, provides tools and features to register, organize, monitor and control IoT devices, as well as check security and update firmware. | Cao, provides tools and features to register, organize, monitor and control IoT devices, as well as check security and update firmware. | Medium, provides tools and features to register, organize, monitor, and control IoT devices, but no security testing or firmware updates. | Medium, provides tools and features to register, organize, monitor, and control IoT devices, but no security testing or firmware updates. |
| Analytical capabilities | High, provides services to collect, store, process, and analyze IoT data in real-time or historically, using technologies such as AWS Kinesis, AWS S3, AWS Lambda, and AWS QuickSight. | High, provides services to collect, store, process, and analyze IoT data in real-time or historical, using technologies such as Azure Stream Analytics, Azure Blob Storage, Azure Functions, and Power BI. | High, provides services to collect, store, process and analyze IoT data in real-time or historical, using technologies such as Google Cloud Pub/Sub, Google Cloud Storage, Google Cloud Functions and Google Data Studio. | High, provides services to collect, store, process and analyze data on the status, performance and life of production equipment. IBM Watson IoT uses technologies such as IBM Watson Studio, IBM Watson Assistant, IBM Maximo Asset Management, and IBM Blockchain. |
| Applicability | High, provides services to create, deploy, and operate IoT applications in the cloud or network edge, using technologies such as AWS IoT Greengrass, AWS SageMaker, AWS IoT Events, and AWS IoT Things Graph. | High, provides services to create, deploy, and operate IoT applications in the cloud or network edge, using technologies such as Azure IoT Edge, Azure Machine Learning, Azure IoT Central, and Azure Sphere. | High, provides services to create, deploy, and operate IoT applications in the cloud or network edge, using technologies such as Google Cloud IoT Core, Google Cloud ML Engine, Google Cloud IoT Device Manager, and Google Cloud | High, provides services to create, deploy, and operate IoT applications in the cloud or network edge, using technologies such as IBM Watson Studio, IBM Watson Assistant, IBM Maximo Asset Management, and IBM Blockchain. |
| Integration capabilities | High, provides APIs and SDKs to connect and interact with other AWS or third-party services and solutions. | High, provides APIs and SDKs to connect and interact with other Azure or third-party services and solutions. | High, provides APIs and SDKs to connect and interact with other Google Cloud or third-party services and solutions. | High, provides APIs and SDKs to connect and interact with other IBM Cloud or third-party services and solutions. |

Table : The pros and cons of IoT framework

#### **Real-life examples illustrate each frame.**

Some practical examples for each IoT framework are:

* **AWS IoT**: An example of AWS IoT is a smart energy management system for buildings. IoT devices such as temperature sensors, light switches, air conditioners, and generators are connected to AWS IoT Core via MQTT. AWS IoT Device Management is used to manage IoT devices remotely. AWS IoT Analytics is used to collect, process, and analyze data about the energy consumption of buildings. AWS IoT Greengrass is used to process data at the network edge and optimize control of IoT devices. AWS SageMaker is used to train and deploy machine learning models to predict and reduce the energy consumption of buildings.
* **Azure IoT**: An example of Azure IoT is a disease monitoring and prevention system for livestock. IoT devices such as temperature sensors, GPS sensors, motion sensors, and RFID scanners are connected to Azure IoT Hub via MQTT or AMQP. Azure IoT Central is used to manage IoT devices remotely. Azure Stream Analytics is used to collect, process, and analyze data about livestock health, location, and activity. Azure IoT Edge is used to process data at the network edge and send alerts when disease risks are detected. Azure Machine Learning is used to train and deploy machine learning models to detect and prevent infectious diseases in livestock.
* **Google Cloud IoT**: An example of Google Cloud IoT is a smart traffic management system for cities. IoT devices such as cameras, pressure sensors, light sensors, and traffic lights are connected to Google Cloud IoT Core via MQTT or HTTP. Google Cloud IoT Device Manager is used to manage IoT devices remotely. Google Cloud Pub/Sub is used to collect, process and analyze data about traffic conditions of the streets. Google Cloud Functions is used to process data at the network edge and control IoT devices in real time. Google Cloud ML Engine is used to train and deploy machine learning models to predict and optimize traffic flows for cities.
* **IBM Watson IoT**: An example of IBM Watson IoT is a smart manufacturing management system for factories. IoT devices such as temperature sensors, vibration sensors, pressure sensors, and industrial robots are connected to IBM Watson IoT Platform via MQTT or HTTP. IBM Watson Studio is used to manage IoT devices remotely. IBM Maximo Asset Management is used to collect, process, and analyze data about the status, performance, and lifespan of manufacturing equipment

### **Standard Tools, How Arduino/Raspberry Pi boards and terminals connect and communicate.**

#### **Tools to support IoT device programming and development.**

Standard tools such as Arduino IDE, Visual Studio Code, Eclipse IoT, Node-RED and PlatformIO are software that support the development and programming of IoT devices.

Arduino IDE is the official integrated development environment for Arduino boards. It allows you to write, compile, and upload code to your Arduino board using a simple graphical interface. It also provides a serial monitor, a library manager, and a board manager. Arduino IDE is easy to use, supports many Arduino models and shields, and has a large community of users and developers. However, Arduino IDE is not very advanced, does not support debugging or code completion, and may not be compatible with some third-party libraries or boards. Some real-life examples of using Arduino IDE are: creating a smart home system with Arduino and sensors¹, building a robot arm controlled by Arduino, and making a weather station with Arduino and LCD display.

Visual Studio Code is a popular code editor that can be used for various programming languages and platforms, including Arduino and Raspberry Pi. It offers many features such as syntax highlighting, code completion, debugging, extensions, and terminal integration. Visual Studio Code is fast, powerful, customizable, and cross-platform. It can be used with PlatformIO or other extensions to enhance the development experience for Arduino and Raspberry Pi. However, Visual Studio Code may have a steep learning curve, require additional configuration and installation steps, and consume more resources than other editors. Some real-life examples of using Visual Studio Code are: developing an IoT application with Azure IoT Hub and Raspberry Pi, programming an Arduino board with PlatformIO, and creating a web server with Raspberry Pi and Node.js.

Eclipse IoT is an open-source community that provides a set of technologies and projects for IoT development. It includes frameworks, tools, protocols, and platforms that can be used to build IoT solutions for various domains and scenarios. Eclipse IoT supports many IoT standards and protocols, such as MQTT, CoAP, LWM2M, OPC UA, etc. It also offers Eclipse Kura, a Java-based framework for building IoT gateways, and Eclipse Ditto, a framework for managing digital twins. Eclipse IoT is comprehensive, flexible, and interoperable. It can be used with Eclipse IDE or other IDEs to develop IoT applications for Arduino and Raspberry Pi. However, Eclipse IoT may be complex, fragmented, and not well documented. Some real-life examples of using Eclipse IoT are: connecting an Arduino board to Eclipse Kura via serial port, implementing an OPC UA server on Raspberry Pi with Eclipse Milo, and creating a digital twin of a Raspberry Pi sensor with Eclipse Ditto.

Node-RED is a low-code programming tool for wiring together hardware devices, APIs, and online services in new and interesting ways. It provides a browser-based editor that makes it easy to create flows using the wide range of nodes in the palette. Node-RED can be run on Raspberry Pi or other platforms, and can communicate with Arduino boards via serial port or other methods. Node-RED is intuitive, expressive, and extensible. It can be used to create IoT applications without writing much code, and to integrate various services and components. However, Node-RED may have limitations in performance, security, and scalability. Some real-life examples of using Node-RED are: controlling an LED on Arduino board with Node-RED dashboard[^10^], reading data from Raspberry Pi sensors with Node-RED nodes, and sending data from Node-RED to Google Sheets.

PlatformIO is a cross-platform, cross-architecture, multiple framework tool for embedded software development. It supports many platforms such as Arduino, ESP32, PIC32, AVR, etc., and many frameworks such as Arduino Core, ESP-IDF, FreeRTOS, etc. It also provides a unified debugger, a unit testing tool, a static code analyzer, and a library manager. PlatformIO can be used with VS Code or other editors to provide a powerful set of tools to assist you in development. PlatformIO is versatile, professional, and modern. It can be used to develop software platforms for various devices and scenarios. However, PlatformIO may be overwhelming for beginners or hobbyists who just want to use the Arduino IDE or similar tools. Some real-life examples of using PlatformIO are: developing an ESP32-CAM project with PlatformIO, creating an IoT application with PlatformIO Home, and building an Arduino Nano project with PlatformIO Core.

#### **Arduino/Raspberry Pi board and end device.**

Arduino/Raspberry Pi board and end device are two popular types of boards used in IoT projects. Arduino is an open source microcontroller board that can be programmed to control devices, sensors, and motors. Raspberry Pi is an embedded computer board that can run the Linux operating system and perform complex computing tasks. End device is a final IoT device that can communicate with other devices through a network. Arduino/Raspberry Pi boards are two popular types of boards used in IoT projects. IoT stands for Internet of Things, a network of devices and objects connected to each other through cloud technology.

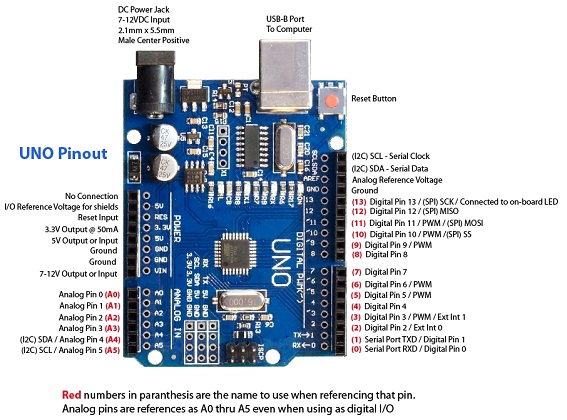
* Arduino is an open source microcontroller board that can be programmed to control devices, sensors, and motors. Arduino comes in different types, for example Arduino Uno, Arduino Nano, Arduino Mega and many more. Each type has different sizes, number of pins, memory and features.

Figure : Arduino Uno

* Raspberry Pi is an embedded computer board that can run the Linux operating system and perform complex computing tasks. Raspberry Pi also comes in many different types, such as Raspberry Pi 3B+, Raspberry Pi 4, Raspberry Pi Zero and more. Each type has different processing speed, RAM capacity, wireless connectivity and features.
* End device is a final IoT device that can communicate with other devices through the network. The end device can be a sensor, a motor, an LED, a push button or any device that can send or receive data.

#### **How to connect and communicate between Arduino and Raspberry Pi boards and terminals**

Arduino and Raspberry Pi can connect and communicate with each other via serial communication. Serial communication is a way to transmit data in series, one bit at a time. This allows the two boards to exchange information with each other. There are two main connection methods: via USB and via GPIOs.

1. Via USB

This method is the simplest, just connect the Arduino to the Raspberry Pi using a suitable USB cable. On Raspberry Pi, you can use any USB port. On the Arduino, you will use the USB port that you use to upload code from your computer (with the Arduino IDE) to the board.

* Advantages: Easy to set up and use, requires no additional components, can use Python libraries such as pyFirmata or pySerial to communicate.
* Disadvantages: Cannot connect multiple Arduinos to one Raspberry Pi via USB, may encounter conflicts with other USB devices, cannot use Arduino's Rx/Tx pins for other purposes⁸.
* For example: Simple IoT projects require data transfer between Arduino and Raspberry Pi, such as LED lights, temperature sensors, or LCD screens.

1. Via GPIOs

This method requires using the Rx/Tx pins of both boards to create serial communication. You also need a 10kΩ pull-up resistor to connect the Raspberry Pi's Rx pin to the 3.3V source.

* Advantages: Can connect multiple Arduinos to a Raspberry Pi via GPIOs, no conflicts with other USB devices, can use Python libraries like pySerial to communicate.
* Disadvantages: More difficult to set up and use, requires additional components, must pay attention to the different voltage between Arduino (5V) and Raspberry Pi (3.3V).
* For example: Complex IoT projects require controlling multiple devices, sensors and motors with Arduino and Raspberry Pi, such as self-driving cars, embedded computers, or smart robots.

### **Sensor.**

Standard sensors for IoT are devices capable of detecting physical or chemical quantities and converting them into electrical signals that can be read and processed by IoT devices. These sensors play an important role in collecting data from the environment and providing information for IoT applications in many fields such as agriculture, healthcare, industry, security, etc. There are many standard sensor types for IoT, depending on the type of quantity that needs to be detected. Here are some common types of sensors and some examples of them:

1. **Temperature sensor**: This sensor measures the heat or coldness of an object or environment. There are different types of temperature sensors such as thermistor, thermocouple, RTD, etc. Applications of temperature sensors in IoT include HVAC control, health monitoring, cold chain management, and more.
2. **Humidity sensor**: This sensor measures the amount of water vapor in the air or in another substance. There are different types of humidity sensors such as resistive, capacitive, optical, etc. Applications of humidity sensors in IoT include smart farming, weather forecasting, forest fire prevention, and more.
3. **Pressure sensor**: This sensor measures the pressure of a liquid or gas on a surface or area. There are different types of pressure sensors such as piezoresistive, piezoelectric, capacitive, etc. Applications of pressure sensors in IoT include blood pressure monitoring, water level measurement, drone control, and more.
4. **Gas sensor**: This sensor measures the concentration of one or more gases in the air or in another substance. There are different types of gas sensors such as electrochemical, optical, semiconductor, etc. Applications of gas sensors in IoT include gas leak detection, air quality monitoring, breath analysis, and more.
5. **Light sensor**: This sensor measures the intensity or wavelength of light in a certain spectrum. There are different types of light sensors such as photodiode, phototransistor, LDR, etc. Applications of light sensors in IoT include smart lighting control, optical fingerprint recognition, sunlight intensity measurement, etc.
6. **Sound sensor**: This sensor measures the volume or frequency of sound within a certain range. There are different types of sound sensors such as microphone, piezoelectric, ultrasonic, etc. Applications of audio sensors in IoT include voice recognition, noise detection, ultrasonic distance measurement, and more.
7. **Vibration sensor**: This sensor measures the vibration or movement of an object or environment. There are different types of vibration sensors such as piezoelectric, piezoresistive, capacitive, etc. Applications of vibration sensors in IoT include earthquake detection, machinery monitoring, physical activity tracking, and more.
8. **Air flow sensor**: This sensor measures the amount of air flowing through a channel or duct. There are different types of air flow sensors such as thermoelectric, optical, electromagnetic, etc. Applications of air flow sensors in IoT include HVAC management, respiratory monitoring, engine control, and more.
9. **Water flow sensor**: This sensor measures the amount of water flowing through a channel or pipe. There are different types of water flow sensors such as vane, spiral, ultrasonic, etc. Applications of water flow sensors in IoT include water consumption management, water leak detection, irrigation control, etc.
10. **Speed sensor**: This sensor measures the speed of an object or environment. There are different types of speed sensors such as optical, electromagnetic, Doppler, etc. Applications of speed sensors in IoT include traffic monitoring, vehicle speed control, human movement tracking, etc.
11. **Acceleration sensor**: This sensor measures the acceleration of an object or environment. Acceleration is a change in the speed or direction of motion. There are different types of accelerometers such as piezoelectric, piezoresistive, capacitive, etc. Applications of accelerometers in IoT include vehicle collision detection, sports and health monitoring, free fall and mobile device collision detection, etc.
12. **Proximity sensor**: This sensor detects the presence of an object near the sensor without physical contact. There are many different types of proximity sensors based on their operating principle and the type of object that needs to be detected.

### **API standard for use in IOT development.**

Standard APIs for IoT are application programming interfaces based on network protocols to enable IoT devices to communicate with each other and with applications. These APIs can come in various forms such as SOAP, REST, XML/JSON, etc. SOAP, REST, XML/JSON are concepts related to standard APIs for IoT. API stands for Application Programming Interface, which is an interface that allows data transfer from one application to other applications via network protocols.

SOAP stands for Simple Object Access Protocol. This is an XML-based communication protocol for exchanging structured information in secure messages. SOAP is a standard maintained by the World Wide Web Consortium and uses protocols like HTTP, SMTP, etc. to transmit messages.

* Advantages: SOAP is highly reliable, supports many types of data, supports user authentication, supports SSL/TLS encryption, compatible with many platforms and applications.
* Disadvantages: SOAP has large message size, consumes bandwidth, does not support resource discovery and observation, and does not support multicast.
* Example: IoT projects require reliable and flexible communication between web applications and IoT devices, such as remote device management, GPS location tracking, or data analysis Iodine.

REST stands for Representational State Transfer. This is an HTTP-based communication architecture for accessing web resources. REST is not a protocol but a set of rules for designing flexible and efficient APIs. REST uses HTTP methods like GET, POST, PUT, DELETE, etc. to make requests and responses.

* Advantages: REST has low latency, saves bandwidth, supports resource discovery and observability, supports SSL/TLS encryption, compatible with many data types such as JSON, XML, etc.
* Disadvantages: REST has low security, does not support user authentication, does not support multicast, depends on the server.
* For example: IoT projects require fast and simple communication between IoT devices, such as smart device control, energy monitoring, or self-driving vehicle management.

XML/JSON are two popular data formats for exchanging information between APIs. XML stands for Extensible Markup Language. This is a markup language for representing structured data in text documents. JSON stands for JavaScript Object Notation. This is a text format for representing JavaScript objects in web applications.

* Advantages: XML/JSON can represent complex data, is easy to read and write, supports many programming languages, and is compatible with many APIs.
* Disadvantages: XML is large in size, difficult to parse, does not support primitive data types such as numbers, strings, etc. JSON is small in size, easy to parse, supports primitive data types, but does not support complex data types like dates, functions, etc.
* For example, IoT projects that require data transfer between APIs can use XML or JSON depending on the data type and performance requirements. XML is often used with SOAP, while JSON is often used with REST.

However, I will focus on APIs based on the following protocols: MQTT, CoAP, HTTP, WebSocket and AMQP.

* **MQTT**

MQTT stands for Message Queuing Telemetry Transport. This is a lightweight and simple serial communication (publish-subscribe) protocol for IoT. MQTT uses an intermediary server (broker) to manage connections and transmit messages between IoT devices (clients). Clients can subscribe or publish messages on different topics.

* Advantages: MQTT is highly reliable, saves bandwidth, is easy to use and expand, supports many levels of QoS (Quality of Service), supports TLS/SSL encryption.
* Disadvantages: MQTT has low security, does not support data storage, does not support user authentication, depends on the broker.
* For example: IoT projects require fast and simple communication between IoT devices, such as smart device control, energy monitoring, or self-driving vehicle management.
* **CoAP**

CoAP stands for Constrained Application Protocol. This is a request-response communication protocol designed for IoT devices with limited capabilities. CoAP uses a client-server model to exchange messages between IoT devices. Messages can be requests or responses from a resource. CoAP uses UDP to transport messages and can be mapped to HTTP.

* Advantages: CoAP has small message size, saves bandwidth, supports resource discovery and observation, supports DTLS encryption, supports multicast.
* Disadvantages: CoAP has low reliability due to the use of UDP, does not support user authentication, and is not compatible with HTTP.
* Example: IoT projects require efficient and secure communication between IoT devices with limited capabilities, such as environmental sensors, smart LEDs, or smart wearables.
* **HTTP**

HTTP stands for Hypertext Transfer Protocol. This is a request-response communication protocol widely used on the web. HTTP uses a client-server model to exchange messages between web applications. The messages can be requests or responses of a web resource. HTTP uses TCP to transmit messages and can use methods such as GET, POST, PUT, DELETE, etc.

* Advantages: HTTP is highly reliable, supports many data types, supports user authentication, supports SSL/TLS encryption, compatible with many platforms and applications.
* Disadvantages: HTTP has large message size, consumes bandwidth, does not support resource discovery and observation, does not support multicast.
* Example: IoT projects require reliable and flexible communication between web applications and IoT devices, such as remote device management, GPS location tracking, or data analysis Iodine.
* **WebSocket**

WebSocket is a full-duplex communication protocol that allows a continuous connection between a client and a server. WebSocket uses a client-server model to exchange messages between web applications and IoT devices. The messages can be requests or responses of a web resource. WebSocket uses TCP to transport messages and can use TLS encryption.

* Advantages: WebSocket has low latency, saves bandwidth, supports full-duplex communication and data push, supports TLS encryption, compatible with HTTP.
* Disadvantages: WebSocket has low security, does not support user authentication, does not support multicast, depends on the server.
* For example: IoT projects require fast and continuous communication between web applications and IoT devices, such as smart chatbots, video calls, or online games.
* **AMQP**

AMQP stands for Advanced Message Queuing Protocol. This is a serial communication protocol (publish-subscribe) that allows one or more producers to send messages to one or more consumers through a queue. AMQP uses a broker to manage connections and transport messages between producers and consumers. Producers and consumers can subscribe to or publish messages on different queues.

* Advantages: AMQP is highly reliable, supports many levels of QoS (Quality of Service), supports user authentication, supports SSL/TLS encryption, compatible with many platforms and applications.
* Disadvantages: AMQP has large message size, consumes bandwidth, does not support resource discovery and observation, does not support multicast, depends on the broker.
* Example: IoT projects require reliable and resilient communication between IoT devices, for example message queue management, event monitoring, or data distribution.

# **Conclusion**

This report has studied different forms of architectures, frameworks, tools, hardware, and APIs for IoT development. The research question of this paper is: Which forms of architecture, frameworks, tools, hardware, and APIs are best suited for different problem-solving requirements in IoT development? To answer this question, this report used a comparison and evaluation method to analyze these forms according to criteria such as security, privacy, compatibility, performance, and cost.

This report has produced the following results and conclusions:

- There is no single form that is optimal for all cases but must depend on factors such as the goals, context, resources, and risks of each IoT project.

- Different forms of IoT architecture have distinct advantages and disadvantages. For example, edge computing has the advantage of reducing latency, increasing security and saving cloud costs, but has the disadvantage of requiring a lot of hardware and high technology; Cloud computing has the advantage of being easy to scale, integrate, and manage, but has the disadvantage of being exposed to security, privacy and latency risks.

- Different forms of IoT frameworks have distinct advantages and disadvantages. For example, AWS IoT has the advantage of having many cloud services, flexibility, and high security, but has the disadvantage of being complex and expensive; Azure IoT has the advantage of having many advanced features, integrating well with Microsoft products, and supporting many programming languages, but has the disadvantage of being difficult to use and unstable.

- Different forms of IoT tools have distinct advantages and disadvantages. For example, the Arduino IDE has the advantages of being easy to use, free, and supports many types of boards, but it has the disadvantages of having few libraries, not supporting debugging, and not being compatible with other platforms; Visual Studio Code has the advantage of having many powerful features, supporting many programming languages and being compatible with other platforms, but has the disadvantage of requiring high configuration and having to install extensions.

- Different forms of IoT hardware have distinct advantages and disadvantages. For example, Raspberry Pi has the advantage of having many functions and is cheap and easy to program, but it has the disadvantage of requiring many accessories, consuming a lot of power and not being able to withstand harsh environments; ESP32 has the advantage of having Wi-Fi and Bluetooth connectivity, saving power and being able to withstand harsh environments, but has the disadvantage of having few functions, being difficult to program and not being compatible with other boards.

- Different forms of IoT APIs have distinct advantages and disadvantages. For example, MQTT has the advantage of being simple, lightweight, and efficient, but has the disadvantage of being insecure and not ensuring reliable delivery; CoAP has the advantages of security, fast response, and reliable delivery, but has the disadvantage of complexity and high configuration requirements.

This paper contributes to the IoT field by providing a general framework to compare and evaluate different forms of architectures, frameworks, tools, hardware, and APIs for IoT development. This report also provides recommendations for IoT developers when choosing the right format for their projects. This report hopes to contribute to the creation of secure, efficient, and cost-effective IoT applications.

# **References**

1. FUNiX Online University (2023) Internet of Things - What is IoT? The simple explanation of IoT. <https://funix.edu.vn/chia-se-kien-thuc/internet-of-things-iot-la-gi-giai-thich-don-gian-ve-iot/>.
2. Contributors to Wikimedia projects (2023) 'Internet of Things,' en.wikipedia.org [Preprint]. <https://en.wikipedia.org/wiki/Internet_V%E1%BA%A1n_V%E1%BA%ADt>.
3. What is IoT? - Explaining the Internet of Things - AWS (2022). <https://aws.amazon.com/vi/what-is/iot/>.
4. Duong T.T.A. (2022) 'What is an IoT device? Popular types of IoT devices today, 'HOT & Fast Real Estate News Updates | Meey Land, 9 June. <https://meeyland.com/chuyen-doi-so/thiet-bi-iot-la-gi-cac-loai-thiet-bi-iot-thong-dung-hien-nay/>.
5. Hoang V. and Hoang V. (2019) 'What is IoT Gateway? What meaning and operating function of this device? ' Blog | TheGioiMayChu, July 31. <https://www.thegioimaychu.vn/blog/tong-hop/iot-gateway-la-gi-y-nghia-va-chuc-nang-van-hanh-cua-thiet-bi-nay-p2134/>.
6. Technology S. (no date) What is IoT? What are the functions and how it work? - Saigon Technology. <https://technolog.vn/tin-tuc/iot-la-gi-chuc-nang-va-cach-thuc-hoat-dong-nhu-the-nao-244.html>.
7. FUNiX Online University (2021) What is the Internet of Things (IoT)? Applications of IoT and Advantages and Disadvantages. <https://funix.edu.vn/hoi-dap-cntt/iot-la-gi-ung-dung-cua-iot-va-uu-diem-nhuoc-diem/>.
8. Tran C. (2022) Internet of Things - What is IoT? <https://securitydaily.net/internet-of-things-iot-la-gi/>.
9. DeepSea (No date) What is IoT and why is IoT important. Available at: <https://www.deepseadev.com/en/blog/why-is-iot-important/>. (Accessed: September 8, 2023).
10. SAS (No date) Internet of Things (IoT): What it is and why it matters. Available at: <https://www.sas.com/en_us/insights/big-data/internet-of-things.html>. (Accessed: September 8, 2023)
11. McKinsey (No date) What is IoT: The Internet of Things explained. Available at: <https://www.mckinsey.com/featured-insights/mckinsey-explainers/what-is-the-internet-of-things>. (Accessed: September 2023).
12. Mojix (2018) Internet of Things: The Importance of IoT in our Everyday Lives. Available at: <https://www.mojix.com/internet-of-things-everyday-lives/>. (Accessed: September 2023).
13. Tateeda (2020) Internet of Things Development: the Good, the Bad and the Future of IoT. Available at: <https://tateeda.com/blog/internet-of-things-the-good-the-bad-and-the-future>. (Accessed: September 10, 2023).
14. Relevant Software (2019) IoT in the Workplace: Benefits, Challenges, Applications and Much More. Available at: <https://relevant.software/blog/iot-in-workplace/>. (Accessed: September 10, 2023).
15. Al-Mutairi and Al-Mutairi (2021) The risks and benefits of Internet of Things (IoT) and their influence on consumer behavior. Available at: <https://www.emerald.com/insight/content/doi/10.1108/SJME-07-2021-0129/full/html>. (Accessed: September 10, 2023).
16. Singh and Singh (2021) Exploring the benefits and challenges of Internet of Things (IoT). Available at: <https://link.springer.com/article/10.1007/s43926-021-00020-9>. (Accessed: September 10, 2023).
17. TesterPro (No date) 10 applications of IoT in modern life. Available at: <https://testerpro.vn/ung-dung-cua-iot/>. (Accessed: 8 September 2023).
18. MeeyLand (No date) What is the Internet of Things (IoT)? Important applications of IoT in life. Available at: <https://meeyland.com/chuyen-doi-so/internet-of-things-iot-la-gi-nhung-ung-dung-quan-trong-cua-iot-trong-cuoc-song/>. (Accessed: 8 September 2023).
19. Kizuna (2019) Smart industrial parks - future investment trends. Available at: <https://www.kizuna.vn/vi/tin-tuc/khu-cong-nghiep-thong-minh-xu-huong-dau-tu-tuong-lai-358>. (Accessed: 8 September 2023).
20. Epicor (2019) What is Industry 4.0—Industrial Internet of Things (IIoT)? Available at: <https://www.epicor.com/vi-vn/resources/articles/what-is-industry-4-0/>. (Accessed: 8 September 2023).
21. NextFarm (2019) SMART AGRICULTURE MODELS IN VIETNAM. Available at: <https://www.nextfarm.vn/cac-mo-hinh-nong-nghiep-thong-minh-o-viet-nam>. (Accessed: 8 September 2023).
22. SmartHomeKit (2019) Suggestions for smart security devices needed in the home in 2023. Available at: <https://smarthomekit.vn> (Accessed: 8 September 2023).
23. EZVIZ (No date) Creating Easy Smart Homes. Available at: <https://www.ezviz.com/vn>. (Accessed: 8 September 2023).
24. WeSmart (2019) 7 smart solutions to ensure home security. Available at: <https://wesmart.vn/tin-tuc/7-giai-phap-thong-minh-de-dam-bao-an-ninh-cho-ngoi-nha.html>. (Accessed: 8 September 2023).
25. Wikipedia (2021) Edge computing. Available at: <https://en.wikipedia.org/wiki/Edge_computing>. (Accessed: 8 September 2023).
26. AWS (No date) What is Edge Computing? - Edge Computing Explained. Available at: <https://aws.amazon.com/what-is/edge-computing/>. (Accessed: 8 September 2023).
27. Microsoft Azure (No date) What Is Edge Computing? Available at: <https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-edge-computing/>. (Accessed: 8 September 2023)
28. IBM (No date) Internet of Things. Available at: <https://internetofthings.ibmcloud.com/> (Accessed: 8 September 2023).
29. Google Cloud (2020) What is Cloud IoT Core? Available at: <https://cloud.google.com/blog/topics/developers-practitioners/what-cloud-iot-core> (Accessed: 8 September 2023).
30. Microsoft Azure (No date) Internet of Things. Available at: <https://azure.microsoft.com/en-us/solutions/iot/> (Accessed: 8 September 2023).
31. AWS (No date) Internet of Things. Available at: <https://aws.amazon.com/iot/> (Accessed: 8 September 2023).
32. AWS Documentation (2021) What is AWS IoT? Available at: <https://docs.aws.amazon.com/iot/latest/developerguide/what-is-aws-iot.html> (Accessed: 8 September 2023).