Carleton University

Systems and Computer Engineering Department
Course Name – SYSC 5809
FALL 2024

Trends in IoT Hardware and Software Platforms

Project Report

SYSC 5809: The Internet of Things

Instructor: Prof. Mohamed Ibn Kahla

04/12/2024

Group 4

Report by:

Mir Md. Saym	101298789
Md Asef Jawad	101326118

Abstract

The rapid evolution of IoT hardware and software platforms is revolutionizing connectivity and data management across industries. On the hardware front, trends emphasize miniaturization, enhanced processing capabilities, and energy efficiency, driven by advancements in microcontrollers, sensors, and edge devices. These innovations enable seamless integration with IoT ecosystems while addressing constraints such as power consumption and cost-effectiveness. Edge computing empowers IoT devices to process data locally that reduces latency and bandwidth demands.

Software platforms are evolving to provide robust, scalable, and secure environments for IoT applications. They focus on interoperability through support for diverse protocols like MQTT and CoAP, real-time analytics, and AI integration. These platforms also enhance user interaction with features such as intuitive dashboards, predictive maintenance capabilities, and real-time decision-making systems. Innovations like containerized microservices and cloud-edge convergence further optimize the deployment and scalability of IoT solutions.

Together, these advancements in IoT hardware and software enable transformative applications in domains such as smart cities, healthcare, and industrial automation. By addressing challenges such as resource constraints, interoperability, and security, these platforms create opportunities for innovation and efficiency. This paper delves into emerging trends, current applications, and future possibilities, highlighting the pivotal role of IoT in shaping a connected and intelligent world. This report also provides a comprehensive analysis of the background and emerging trends in IoT hardware and software, with challenges and opportunities for future innovation across multiple sectors.

Acknowledgements

We express our deepest gratitude to our instructor and mentor, whose guidance, constructive feedback, and technical expertise played an important role in shaping the direction of this project. His support and encouragement have been instrumental in overcoming challenges and refining the focus of the study on IoT hardware and software platforms.

We also thank the institutions and open-source communities that provided access to resources and data, particularly IEEE, ACM, Elsevier, and a few other journals and websites. The collaborative efforts of our peers (the student audience) and the stimulating discussions during this process have enriched the study, which allowed a comprehensive exploration of this dynamic field. Finally, we thank each other as group members for our collaborative effort in shaping the project.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

IoT	Internet of Things
NB-IoT	Narrowband Internet of Things
AWS	Amazon Web Services
MQTT	Message Queuing Telemetry Transport
CoAP	Constrained Application Protocol
IDE	Integrated Development Environments
RAM	Random Access Memory
PWM	Pulse Width Modulation
NFC	Near Field Communication
LoRaWAN	Long Range Wide Area Network
LPWAN	Low Power Wide Area Network
LTE / 5G	Long Term Evolution / Fifth Generation
SoC	System on Chip
TPU	Tensor Processing Unit
M2M / D2D	Machine to Machine / Device to Device
NETCONF	Network Configuration Protocol
AMQP	Advanced Message Queuing Protocol
GDPR	General Data Protection Regulation
PCI-DSS	Payment Card Industry Data Security Standard
DDS	Data Distribution Service
HTTP	Hypertext Transfer Protocol
STOMP	Simple Text Oriented Messaging Protocol
RDS	Relational Database Service
WS EMS	WebSocket-based Edge MicroServer
API	Application Programming Interface
SDK	Software Development Kit
OPC-UA	Open Platform Communications Unified Architecture
XMPP	Extensible Messaging and Presence Protocol
MDR	Metadata Registry
SDN	Software-Defined Networking

1. Introduction

The Internet of Things (IoT) has become a transformative force across various industries, connecting devices and systems to enable smart environments and advanced data-driven solutions. IoT platforms play a central role in this global ecosystem by managing device connectivity, data processing, and system integration. These platforms provide the infrastructure and tools necessary to build and deploy IoT applications across domains such as smart cities, healthcare, industrial automation, and environmental monitoring.

1.1. Types of Platforms

The landscape of IoT platforms is vast and varied. It is designed to meet the various requirements of modern applications. Each platform is designed to address specific challenges, such as device integration, data management, or application development. These platforms can be broadly categorized based on their primary components and the accessibility of their source code.

1.1.1. Platforms based on Components

IoT platforms can be categorized into hardware, software and middleware-oriented systems based on their primary functions [1] [2].

Hardware Platforms: Focus on physical components like microcontrollers, sensors, and actuators to provide connectivity and environmental monitoring.

Software Platforms: Provide tools for data management, analysis, and visualization, including cloud services, mobile applications, and IDEs.

Middleware Platforms: They act as intermediary platforms, provide communication, data processing, protocol translation, and integration across IoT systems.

1.1.2. Platforms based on Sources

IoT platforms can also be categorized based on the accessibility of their source code [1].

Open-Source Platforms: Provide public access to source code ensuring flexibility and transparency.

Closed-Source Platforms: Proprietary systems with restricted access, typically supported by commercial providers for enhanced stability and security.

1.2. Contribution and Organization

This report provides a comprehensive analysis of IoT platforms, covering their components, advancements, and applications. We have organized the report as follows:

Chapter 1 provides an overview of IoT platforms, categorizing them based on components and source accessibility, and their role in connectivity, data processing, and integration.

Chapter 2 focuses on IoT hardware platforms, their core components like sensors, actuators, and microcontrollers, as well as advancements in hardware efficiency and scalability.

Chapter 3 explores IoT software and middleware platforms, highlighting their architectural requirements, management perspectives, and integration of data protocols and standards.

Chapter 4 compares popular IoT platforms, analyzing their strengths, compatibility, and protocols, with comparisons of open and closed-source platforms.

Chapter 5 provides current trends in IoT platforms, including edge-cloud computing, cross-platform compatibility, lightweight operating systems, and AI integration and 5G networks.

Chapter 6 presents some applications and use cases of IoT platforms, such as digital twins of wind turbine, smart city platforms, and smart farm management systems.

Chapter 7 examines challenges in IoT, including security, scalability, interoperability, and energy efficiency, while proposing future solutions.

Chapter 8 concludes with a summary of the report, emphasizing the transformative potential of IoT platforms in modern industries.

2. IoT Hardware Platforms

2.1. Core Hardware Components

2.2. Sensors

Sensors act as the primary interface between IoT systems and the physical environment. They detect and measure various physical phenomena such as temperature, humidity, motion, and light. Recent advances in IoT have introduced multifunctional sensors (e.g. gas and tactile sensors) that can detect multiple parameters simultaneously which reduce the complexity and cost of IoT systems. As an example, an environmental sensor is capable of monitoring temperature, humidity, air pressure, and air quality simultaneously [3]. The following list outlines some of the popular IoT sensors.

Figure 1. Different types of sensors for Iot applications.

2.2.1. Actuators

Actuators allow IoT devices to interact physically with the environment by transforming electrical signals into mechanical actions. Common examples include servos for precise motor control, solenoids for locking mechanisms, and relays for switching high-power circuits. Actuators are commonly paired with sensors to build responsive systems, such as irrigation systems that adjust water flow based on soil moisture readings [3]. Based on their construction mode and role in the IoT environment, they can be divided into four types as shown in Figure 2.

i. ***Linear actuators***: They are used to make objects or elements move in a straight line.
ii. ***Motors***: They make precise rotational movements of equipment components or objects.
iii. ***Relays***: This type includes electromagnetic-based actuators used to operate lights, heaters and even power switches in smart cars.
iv. ***Solenoid valves***: Most widely used in household appliances as part of locking mechanisms.

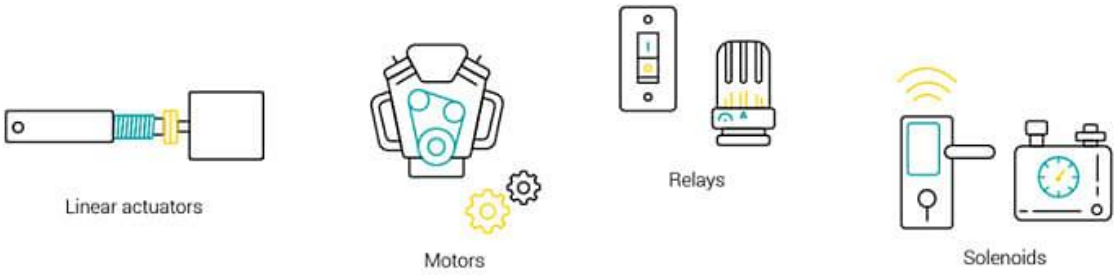

Figure 2. Different types of actuators for IoT applications [4].

2.2.2. Microcontroller

Microcontrollers serve as the computational core or brains of IoT devices, managing data processing and device coordination. They execute instructions from embedded programs to process data received from sensors and control actuators. Widely used platforms such as Arduino, ESP32, and Raspberry Pi have significantly lowered the entry barrier for IoT development by offering programmable interfaces, broad compatibility, and community-driven support [1][4].

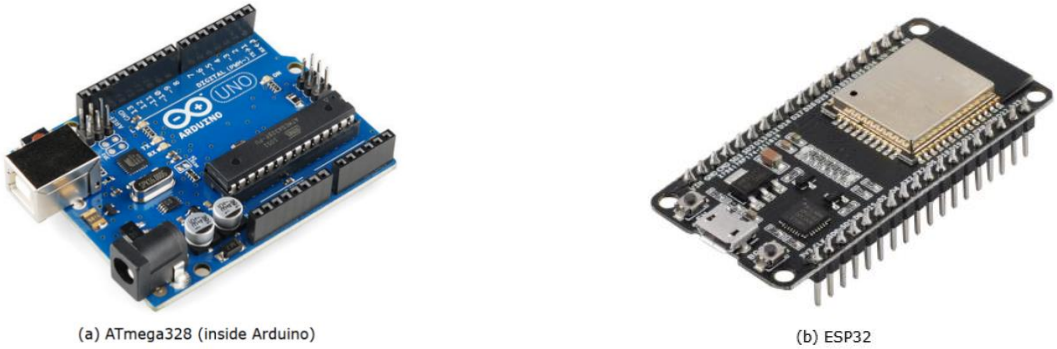

Figure 3. Microcontroller units – ATmega 328 and ESP32.

The ATmega328 is an 8-bit AVR RISC-based microcontroller commonly found in Arduino boards, making it a staple for IoT development. It offers 32 KB of Flash memory for program storage, 2 KB of SRAM for runtime data, and 1 KB of EEPROM for non-volatile data storage. The ATmega328 supports key communication protocols, including UART, I2C, and SPI, essential for connecting sensors and actuators [4]. Moving to ESP32 microcontroller, it is a powerful 32-bit dual-core processor widely recognized for its built-in Wi-Fi and Bluetooth capabilities. With 520 KB of SRAM

and support for external flash memory, the ESP32 can handle computationally intensive tasks and manage large dataset [5].

2.3. Reference Hardware Platform – Arduino UNO

Arduino boards utilizing the ATmega328 microcontroller, are designed to provide simplicity and efficiency. These boards implement the AVR architecture based on the Harvard model, which segregates program code and data into separate memory spaces, optimizes memory allocation and enhances processing performance. Arduino UNO boards include 14 digital I/O pins, with 6 supporting PWM functionality, which is needed for controlling devices such as motors and LEDs with precision. Power supply options include USB or external sources (7–12V), supported by an integrated voltage regulator for stable and reliable performance [4][5].

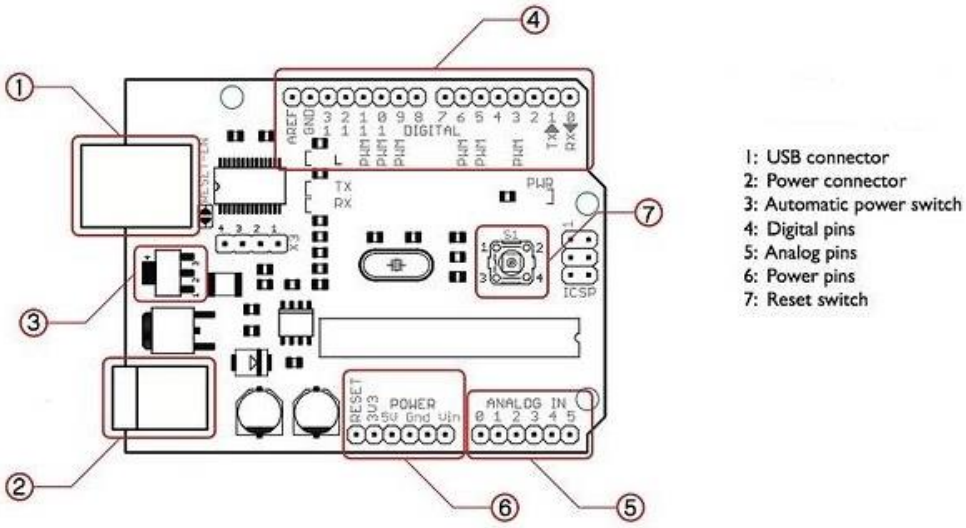

Figure 4. Basic board Diagram of Arduino UNO.

- ***USB Connector*** – To connect the Arduino to a computer for programming and power supply.
- ***Power Connector*** - Allows external power sources (e.g., battery) to power the Arduino board.
- ***Automatic Power Switch*** - Automatically switches between USB power and external power.
- ***Digital Pins (0-13)*** - Can be configured as input or output for reading and sending signals.
- ***Analog Pins (A0-A5)*** - Used to read analog signals and convert them to digital form.
- ***Power Pins*** - Provide or receive power to/from the board, including 5V, 3.3V, GND, and VIN.
- ***Reset Switch*** - Resets the microcontroller, restarting the program from the beginning.

2.4. Communication Standards

Connectivity is a fundamental requirement for IoT systems. We briefly summarise some of the most widely used communication protocols within IoT, their key features, operational modes, technical characteristics and typical use cases for each protocol (Table. 1).

Table 1. Common communication protocols for IoT system.

Communication Protocol	Data Rate	Range	Frequency Band
NFC	424 kbps	20 cm	13.56 MHz
Bluetooth	2.1 Mbps	100 m	2.4 GHz
Bluetooth Low Energy	0.27 Mbps	100 m	2.4 GHz
Wi-Fi (802.11ac)	1.3 Gbps	50–70 m	5 GHz (2.4 GHz in earlier versions)
ZigBee (802.15.4)	20–250 kbps	10–20 m	2.4 GHz, 900 MHz, 868 MHz
Z-Wave	40 kbps	100 m	800–900 MHz
6LoWPAN (802.15.4)	20–250 kbps	10–20 m	2.4 GHz, 900 MHz, 868 MHz

2.5. Edge Devices and Gateway

Edge devices, such as routers, switches, servers, and embedded IoT devices, collect and process data close to the source in real-time and reduce latency. Gateways act as intermediaries, aggregating data from multiple devices, perform initial processing, and ensure secure transmission to cloud platforms. This combination optimizes bandwidth usage, enhances security, and improves response times for IoT systems.

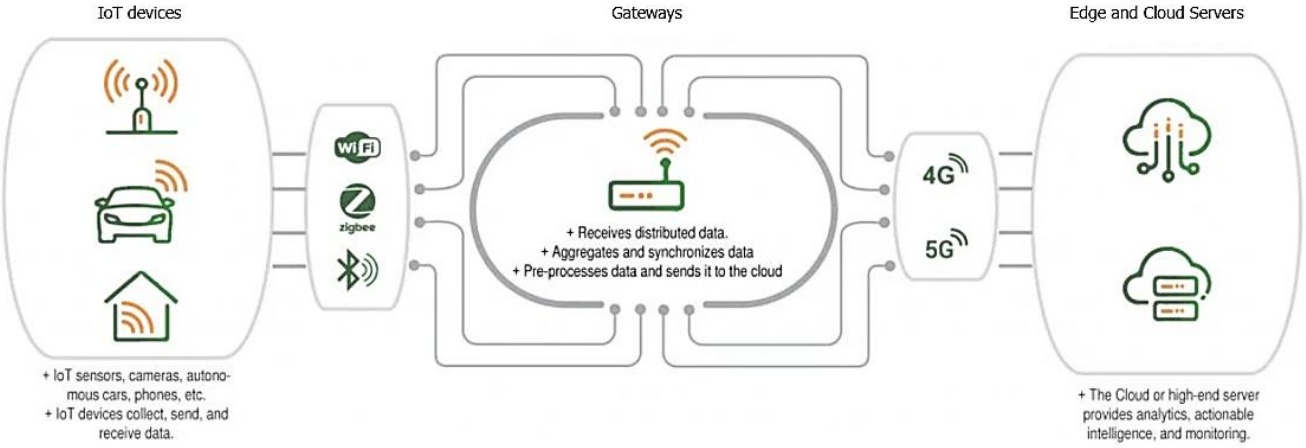

Figure 5: Simplified Edge devices and gateway interconnectivity.

Edge Routers: Routers are networking devices that provide routing services between local IoT devices and external networks, such as the internet or a cloud service. They manage data traffic, perform protocol conversion, and implement security measures like firewalls and VPNs [9].

Edge Switches: These switches manage communication between multiple devices in a local network by forwarding data packets. In IoT systems, they typically connect IoT devices to gateways or other edge infrastructure [9].

Edge Servers: They provide compute resources at the edge of the network, processing and analyzing data from IoT devices locally before sending it to the cloud. This reduces latency and bandwidth consumption. An example is an industrial edge server that collects real-time data from IoT sensors, analyzes it locally, and sends only critical information to the cloud [10].

IoT Hubs/Controllers: These devices connect, manage, and control IoT devices in a network, often serving as local control points. Smart home hubs like Amazon Echo or Google Home manage various IoT devices and communicate with the cloud, facilitating smart home and office automation [6].

Gateways: Protocol gateways translate data between different IoT communication protocols, enabling interoperability, such as converting Modbus to MQTT for industrial IoT systems. LPWAN gateways connect low-power devices over long distances using technologies like LoRa, Sigfox, or NB-IoT, ideal for sensors and smart meters. Mesh gateways extend network coverage by relaying data across devices in a mesh network, enabling communication even outside direct hub range [10][11].

2.6. Advancement in Hardware

Advancements in Hardware The evolution of IoT Hardware has been driven by many key factors like size reduction, power efficiency and processing capabilities. These developments support the growing demands of IoT applications in industries such as healthcare, agriculture, and smart cities.

1. ***Size Reduction:*** The size reduction of IoT hardware devices is a significant trend. Nowadays, chips can be deployed in constrained spaces such as wearable devices, implants, and compact sensors. This is achieved through advancements like System-on-Chip (SoC) technology, which integrates multiple functionalities onto a single chip [12].

2. ***Power efficiency:*** In IoT devices, it is achieved using ultra-low-power processors and energy harvesting techniques. Modern IoT devices often incorporate ultra-low-power microcontrollers, such as the ARM Cortex-M series, to minimize energy consumption during operation [6].
3. ***Improved Processing***: Advancements in IoT hardware have enabled the integration of edge computing, allowing real-time data analysis directly on devices and reducing reliance on cloud processing [10]. AI-driven chips, such as the NVIDIA Jetson Nano and Google Edge TPU, include accelerators for efficient on-device processing of AI and machine learning tasks [12].

3. IoT Software & Middleware Platforms

IoT software and middleware platforms support the necessary infrastructure to ensure interoperability, scalability, and security across IoT systems, which is essential for efficient and effective IoT deployments [1][13].

3.1. Reference IoT Platform Model and Requirements

Most IoT cloud platforms provide a connectivity layer service that enables seamless interaction between edge devices and the platform. Therefore, these platforms typically focus on specific aspects such as connectivity management, device management, and, in some cases, data visualization [14]. Figure 6 illustrates the key component requirements of an IoT platform, and the following section provides a detailed description of each component. The main functional requirements are as follows:

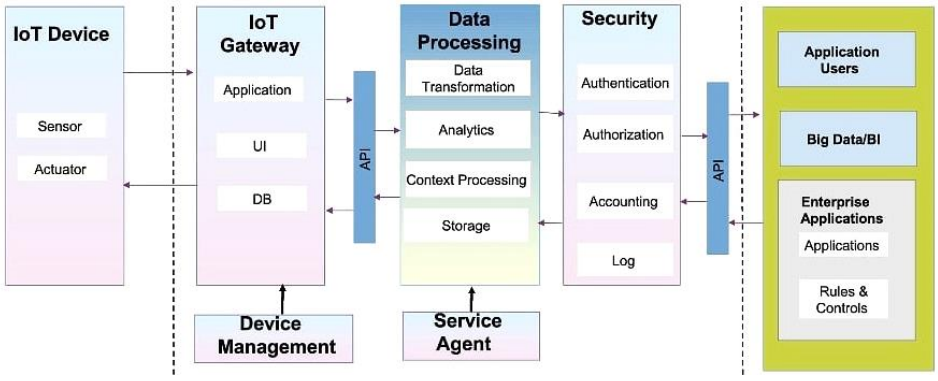

Figure 6. Simplified IoT software/middleware platform model.

IoT platform requirements include ***device management*** for remote configuration using protocols like NETCONF, and ***network management*** to handle scalability and communication protocols such as MQTT and CoAP. ***Platform management*** oversees data processing for decision-making, while ***resource management*** ensures efficient resource discovery, monitoring, and optimization to reduce and costs. ***Security and privacy*** address cyberattacks like denial of service with standards like GDPR and PCI-DSS**.** ***Data management*** ensures organized, accessible data with backups, while ***analytics*** utilizes machine learning and business intelligence for actionable insights. ***External interfaces*** provide APIs, SDKs, and gateways for integration and application development [4][13][14][16].

3.2. Events, Protocols and Standards

In IoT, an event refers to the response of devices to real-time physical changes in their environment, such as a temperature sensor detecting a rise in heat. Event handling occurs at the application level and relies on specific protocols, which typically follow one of two approaches: the *publisher/subscriber* model or the *polling* model [1].

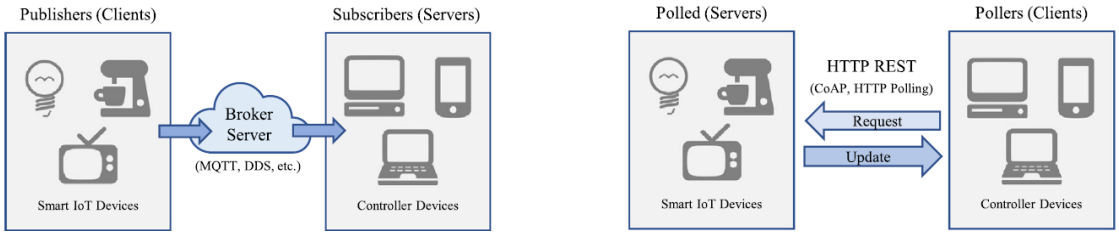

Figure 7. Event handling framework in IoT platform: a) Publisher-Subscriber, b) Polling [1]

Publisher-Subscriber: This approach is a widely used method for event handling in IoT platforms. In this model, an IoT device (the publisher) streams real-time data to designated controller devices (the subscribers), which manage and adjust settings for the devices. A broker server operates as an intermediary [15].

Polling: This is another method for event handling in IoT, following a *client-server* model. In this approach, a Poller device, typically a controller acting as the *client*, periodically requests data from IoT devices, which function as *servers*. Unlike the publisher-subscriber model, the roles are reversed

in the polling model [1][15].

Message Query Telemetry Transport (MQTT): It is a communication protocol commonly used in M2M and IoT systems. An MQTT server, known as the broker, acts as the central hub where publishers and subscribers connect using a shared identifier. In this setup, IoT devices function as publishers, while IoT hubs or control devices serve as subscribers [1].

Advanced Message Queuing Protocol (AMQP): AMQP is an open standard application-level protocol that supports message delivery in three modes: at-most-once, where delivery is not guaranteed; at-least-once, ensuring the message is delivered at least once; and exactly-once, where the message is delivered precisely one time [15].

Data Distribution Service (DDS): This is a protocol designed for machine-to-machine (M2M) communication in real-time IoT applications. Unlike XMPP, DDS ensures quality of service and guarantees message delivery. It does not rely on a centralized broker like MQTT; instead, it uses multicast as the primary mechanism to distribute messages among all devices in the system [1].

Simple Text Oriented Message Protocol (STOMP): It is a lightweight, asynchronous protocol that enables message exchange between entities using a publisher/subscriber model with intermediate servers [1].

Hypertext Transfer Protocol (HTTP): It is the base of the communications for the world wide web. It is a generic stateless protocol which follows the client–server model and can be used in IoT to exchange data with a plain publish model or with a polling model [15].

Constrained Application Protocol (CoAP): It is a polling protocol designed for resource-constrained IoT devices. Based on the REST architecture, CoAP supports multicasting through the UDP protocol. It implements four message types: confirmable, non-confirmable, reset, and acknowledgment [15].

4. Popular IoT Platforms and Comparison

4.1. Closed Source IoT Platform

4.1.1. Azure IoT Core

The Azure IoT is designed to control and manage billions of IoT devices by using Microsoft cloud services. Its architecture revolves around three key modules. The device module collects data from specific environments and sends it to the cloud for processing. Once in the cloud, the data undergoes analysis to generate actionable insights. The action module utilizes these insights to trigger specific outputs or responses. Data analysis is carried out through hot and cold paths, with the hot path processing data in real-time for immediate actions, while the cold path focuses on long-term data analysis [14][16].

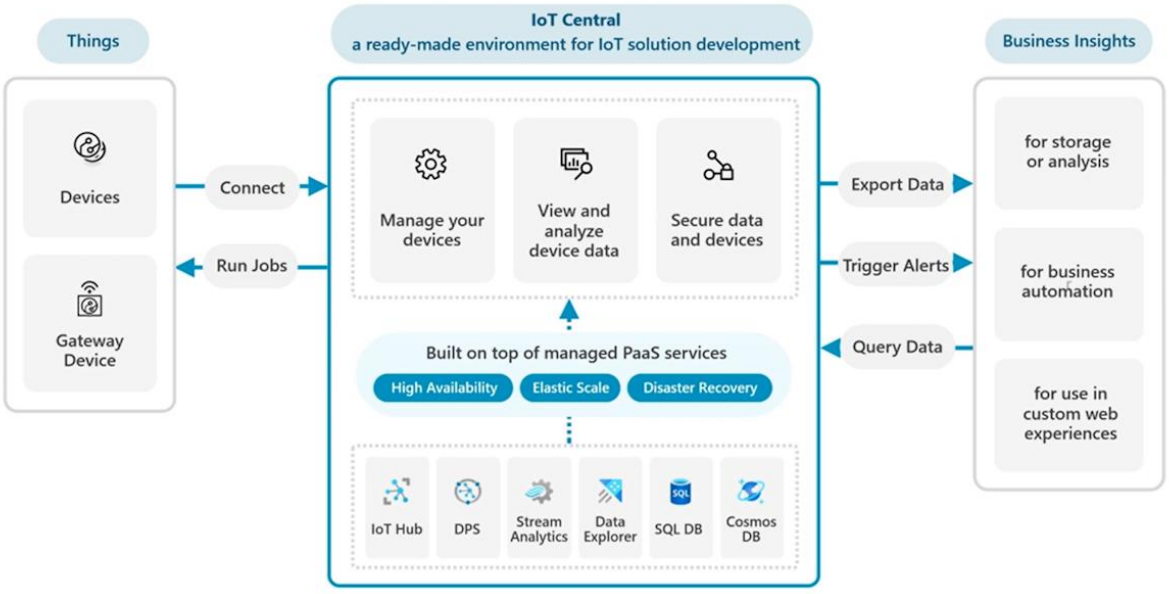

Figure 8. Reference Architecture of Azure IoT core [16].

The Azure gateway connects certain devices to the platform, securely transmitting data to the cloud through the Azure IoT cloud gateway. This gateway handles device management and provides continuous monitoring, supporting reliable full-duplex communication for both client and backend applications. The device provisioning service registers devices and links them to the IoT hub. The business process manages functions such as storing messages and transferring emails [16].

4.1.2. AWS IoT

AWS IoT is a managed cloud service designed for secure and reliable interaction between connected devices and cloud applications or other devices. It integrates with services like AWS Lambda and

AWS Relational Database Service (RDS) to develop IoT applications capable of collecting, processing, analyzing, and responding to data from connected devices [14]. The Device Gateway acts as an intermediary between devices, cloud services, and connected devices, ensuring secure communication through TLS. The Message Broker uses the MQTT protocol with a publish/subscribe pattern for low-latency data exchange between IoT devices and applications. AWS Lambda supports stateless functions triggered by various sources to extend access control mechanisms. AWS RDS provides scalable, cost-efficient database management with automated tasks like provisioning and backups [17].

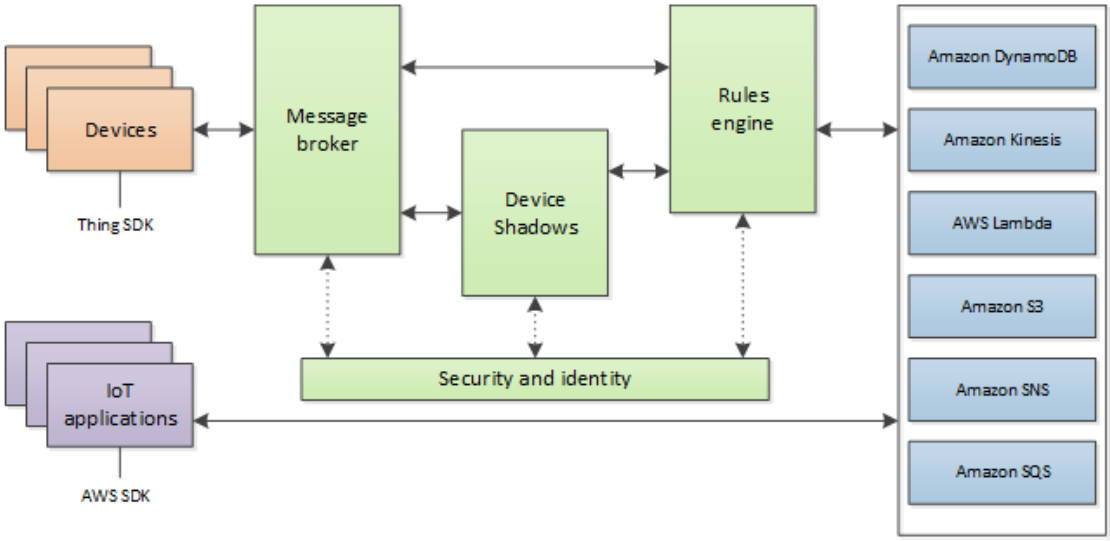

Figure 9. Simplified AWS IoT model.

4.1.3. ThingWorx

ThingWorx is a powerful IoT platform developed for industrial applications with a layered architecture. Its Device Layer supports hardware interactions via WebSocket-based Edge MicroServer (WS EMS) and ThingWorx Edge SDKs, while the Database Layer manages scalable data storage with PostgreSQL and SQL Server [19]. Widely used in manufacturing, energy, and smart cities, it enables asset tracking, predictive maintenance, and operational analytics. Its modular design ensures scalability, integration with cloud and edge systems, and compatibility.

4.2. Open Source IoT Platforms

4.2.1. Fiware

FIWARE is an open-source platform designed for smart application development which focuses on scalability and interoperability. It supports IoT integration via NGSI APIs and protocols like MQTT and CoAP that allows real-time updates and seamless device communication. Applications span smart cities, agriculture, and healthcare, optimizing resources and enhancing efficiency [19]. FIWARE's advantages include cross-domain interoperability, driving economic growth, and ensuring security with *OAuth2* and *XACML* protocols [19].

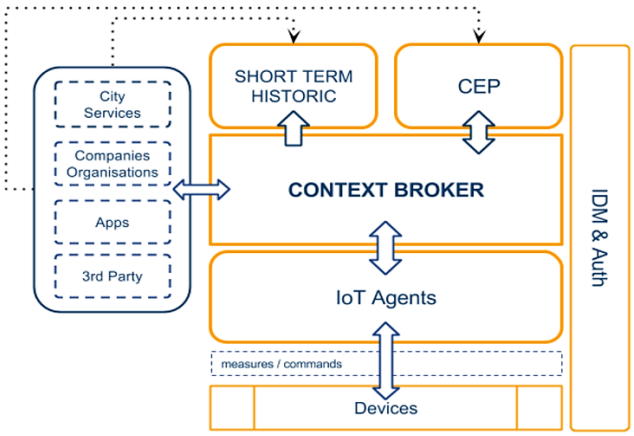

Figure 10. Simplified FIWARE IoT model.

4.2.2. ThingSpeak

ThingSpeak is a cloud-based IoT platform designed for capturing, visualizing, and analyzing real-time data. It integrates seamlessly with MATLAB for advanced analytics and processing. The platform supports key functionalities like data transformation, automated scheduling of calculations, and combining data sources for complex analysis. Applications include smart agriculture, environmental monitoring, and academic research [17].

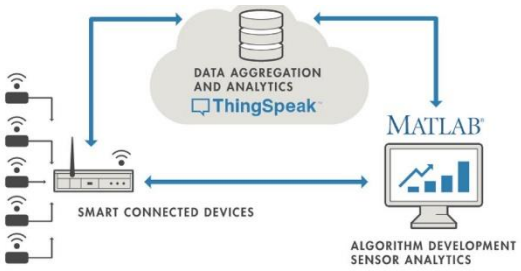

Figure 11. ThinkSpeak IoT model with MATLAB [14].

4.2.3. Eclipse Kapua

Eclipse Kapua is an open-source IoT cloud platform for managing connected devices and processing IoT data, with robust lifecycle management, secure data collection, and remote monitoring. It supports industrial IoT for predictive maintenance, smart farming for real-time environmental monitoring, and smart cities for managing infrastructure like lighting and energy. Its modular design ensures scalability, while its open-source nature promotes innovation. It is cost-effective, eliminating licensing fees, and ensures security with strong protocols for data transmission and device authentication, making it ideal for diverse IoT applications [20].

4.3. Comparison of Leading IoT Platforms

In this section we will mainly compare popular open and close source platforms mentioning their strength, management, protocol support to evaluate their use cases. Table 2 highlights the similarities and differences among the popular IoT platforms. These platforms, including Azure IoT, FIWARE, ThingWorx, and ThingSpeak, AWS IoT, manage devices in distinct ways. For instance, the Azure Cloud offers an IoT Hub that acts as a gateway to enhance data management and provisioning [16]. While most platforms utilize identical protocols, FIWARE stands out by using an additional client-server protocol called Open Platform Communications Unified Architecture (OPC-UA) [18]. Common data protocols across the platforms include MQTT, CoAP, and XMPP. Machine-to-machine support is enabled through Open Mobile Alliance Lightweight Machine-to-Machine (OMA-LWM2M), along with programming languages such as Java, Python, C, and Matlab, enhancing functionality on IoT platforms [14][21]

Table 2. Comparison of popular open and closed sourced IoT platforms [1][14][15][22].

IoT Platforms	Platform Type	Device Management	Connectivity Management	Data Management	Cryptography Algorithm	Server Protocols	Programming Languages
Microsoft Azure IoT	Closed Source	Azure IoT Hub for device provisioning, provide SDK	Wi-Fi, Bluetooth, NFC, Ethernet, IPv6	Provides Event Hub service for data communication with the cloud, data storage	AES, RSA, SHA, DTLS, TLS	HTTPS, MQTT, AMQP	.NET C#, Java, Node.js, Python, iOS, Go, REST API
Amazon AWS IoT	Closed Source	Device provisioning, authentication, fleet management, OTA firmware	Wi-Fi, Ethernet, Bluetooth, Zigbee, LoRaWAN, Cellular, Satellite	Data collection and storage with AWS services (S3, DynamoDB,RDS, IoT Analytics)	AES-128, RSA, TLS	MQTT, HTTPS, WebSockets	Java, Python, Node.js, C, C++, JavaScript, Go
FIWARE	Open Source	NGSI interface for context data management, open-source and interoperable system	Wi-Fi, Bluetooth, cellular, RFID, LoRaWAN, Ethernet, IPv6	NGSI-LD interface, integrates with databases for big data analytics and visualization	TLS, SSL	LWM2M, MQTT, OPC-UA, LoRaWAN	Python, C, C++, Java, Node.js
Ericsson IoT Accelerator	Closed Source	Offer device management, connectivity management, and scalability	Cellular, Wi-Fi, Ethernet	Provides real-time data management and analytics using integrated tools	AES-256, TLS	MQTT, HTTPS, CoAP	Java, Python, C++
ThingWorx	Closed Source	Platform for managing and analyzing data in the industrial IoT	Wi-Fi, NFC, IPv6, RFID, Bluetooth, LoRaWAN	Uses the ThingWorx Kepware server to manage devices	TLS/AES-128	MQTT, XMPP, CoAP, JSON, Sigfox	Java, C, .Net
ThingSpeak	Open Source	Control devices, channel status updates, export rdata	Wi-Fi, Ethernet, Bluetooth, Cellular	Uses the ThingSpeak channel to explore relationships in data	AES-128	HTTP, MQTT	MATLAB

5. Trends in IoT Platforms

5.1. Cloud and Edge Computing

Edge-cloud convergence combines the strengths of edge and cloud computing to meet the demands of AI powered IoT applications. As seen in Figure 12, this reference model establishes a cooperative framework among edge devices, edge servers, and cloud platforms to ensure efficient task distribution and data processing. In this model, edge devices, such as IoT sensors and actuators, gather data and perform basic preprocessing tasks. Edge servers, located near these devices, perform operations requiring low latency to reduce delays and minimize the need to transmit raw data to distant servers. Meanwhile, the cloud platform manages computationally intensive processes like training AI models and performing large-scale data analysis [23].

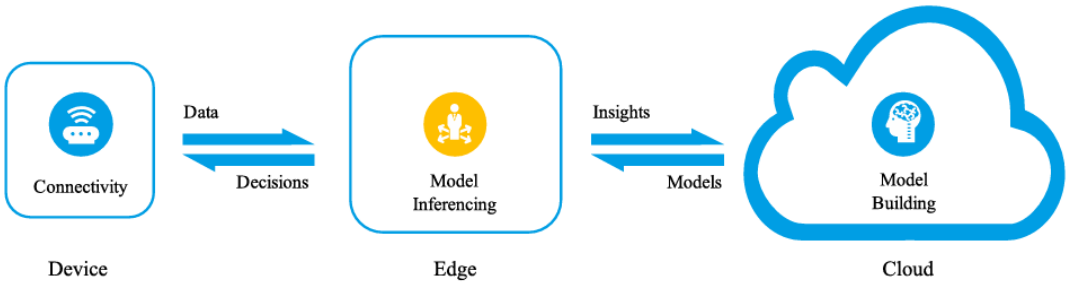

Figure 12. Reference Edge-Cloud convergence model for AI-IoT.

5.2. Cross-Platform Compatibility

The cross-platform access control mechanism can address the challenge of achieving secure and efficient interoperability among different Industrial IoT (IIoT) platforms. Instead of relying on integrated access control systems, it may proceed by mapping and sharing user attribute metadata. Blockchain technology and a metadata registry (MDR) form the core of this approach that can secure exchange of user attributes while maintaining data integrity and traceability. By avoiding the direct sharing of sensitive access control policies, the framework reduces security risks. The key components of the framework include a permissioned blockchain which ensures secure metadata sharing among trusted platforms to enhance reliability. The MDR acts as a centralized repository for

mapping user attributes across platforms. It also supports precise access control without requiring a unified data format. Additionally, this mechanism uses the native authentication and authorization mechanisms of individual platforms that minimizes complexity and operational overhead [24].

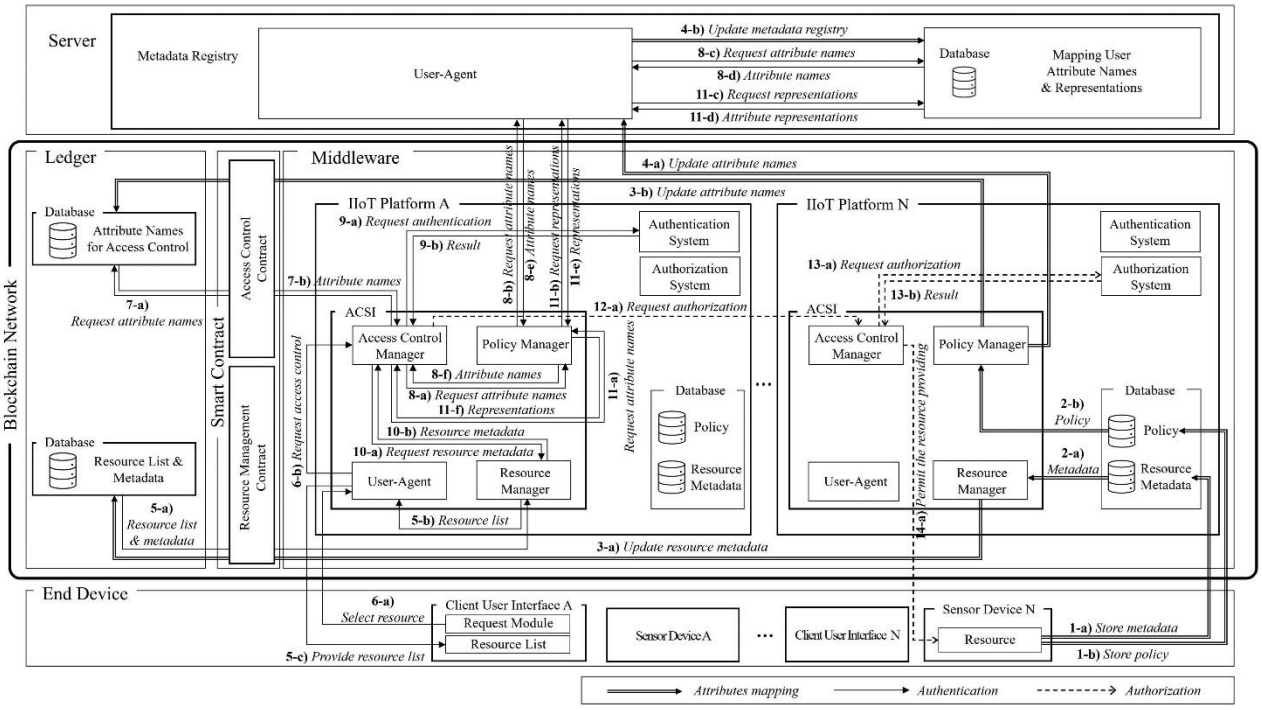

Figure 13. A reference cross platform access control model for IIoT.

5.3. Lightweight Operating Systems

Lightweight OS play an important role in IoT platforms by optimizing resource-constrained devices and ensuring efficient operation. These OS are designed to work effectively with limited memory, processing power, and energy resources that makes them suitable for IoT systems. They manage tasks such as real-time data processing, scheduling, and network communication and ensure low power consumption. Some examples include Contiki, *TinyOS*, *FreeRTOS*, and RIOT, which support features like modular architectures, dynamic memory management, and energy-efficient scheduling. Recent advancements in IoT OS development include support for IPv6 networking, multithreading, and enhanced real-time capabilities. *FreeRTOS* now incorporates tick-less scheduling for reduced power consumption whereas RIOT supports dynamic protocol stacks for greater flexibility [25]. Emerging IoT OS support also integrates security features like secure boot and memory protection.

5.4. AI and Machine Learning

Artificial Intelligence (AI) has been deeply integrated into IoT systems that has advanced automation, decision-making, and efficiency. Applications like Alexa and Google Home use natural language processing. AI-enabled robots like Pepper and Sophia upgrade machine learning and computer vision for human interaction and task execution. AI techniques like data mining and fuzzy logic, enhance resource management and enable real-time system adaptations, like lighting optimization. Future advancements include federated learning for decentralized AI training, quantum-resistant cryptography, and AI-driven dynamic network optimization. These innovations strengthen IoT software by improving security and enabling real-time analytics. For example, neural networks embedded in IoT platforms predict device failures and optimize maintenance and operational efficiency in systems powered by 5G and beyond [26].

5.5. 5G and Future Networks

5G and future cellular networks play an important role in IoT software and middleware platforms by addressing critical challenges such as connectivity, scalability, and latency. Technologies like massive MIMO, carrier aggregation, and device-to-device (D2D) communication provide wider connectivity for heterogeneous networks. This connectivity is essential for IoT platforms managing diverse devices and applications. These allow IoT middleware to manage large-scale data streams efficiently and support real-time analytics and decision-making. The integration of 5G with cloud and edge computing enhances the responsiveness of IoT platforms by distributing processing tasks closer to devices. This is crucial for latency-sensitive applications in sectors like healthcare, smart cities, and industrial automation [5].

Features and support of 5G for ultra-low latency, high throughput, and advanced protocols such as software-defined networking (SDN) and network slicing guarantee quality of service (QoS) for critical IoT applications. These features ensure reliable and secure communication, the development of applications requiring strict performance standards [2]. Middleware platforms also benefit from the enhanced security and privacy mechanisms offered by 5G, such as network isolation and secure

key management.

5.6. Software Security

IoT systems disclose significant vulnerabilities due to their distributed architecture and the resource constraints of connected devices. Notable weaknesses include firmware that can be tampered with, insufficient cryptographic key management, and reliance on insecure software-based cryptographic methods. These issues require the adoption of robust measures such as hardware-based Roots of Trust (RoT) for safeguarding cryptographic keys, secure boot protocols to validate firmware integrity, and cryptographic accelerators to enhance the efficiency and security of encryption processes. ARM's Platform Security Architecture (PSA) strengthens IoT security by implementing TrustZone technology to isolate secure and nonsecure operational domains. It enforces authenticated firmware execution through secure boot mechanisms and incorporates cryptographic accelerators [10]. This framework plays a pivotal role in securing IoT deployments in domains such as smart cities, healthcare, and industrial systems by ensuring reliable device authentication, maintaining data integrity, and enabling encrypted communication.

6. Applications and Use Cases

6.1. Digital Twin of Wind Turbine

The digital twin architecture for the wind turbine on the Azure IoT platform integrates physical turbine operations with virtual simulations for monitoring and analysis. Sensors on the turbine measure parameters such as wind speed, rotational speed, pitch angle, and mechanical power. This data is transmitted securely to the Azure IoT Hub via MQTT. Azure Digital Twins (ADT) serves as the core that creates virtual representations of turbine components and synchronizes real-world telemetry in real time. Processed data is analyzed using Azure Stream Analytics and Machine Learning for anomaly detection, performance optimization, and predictive maintenance. Dashboards provide real-time visual insights for informed decision-making [24].

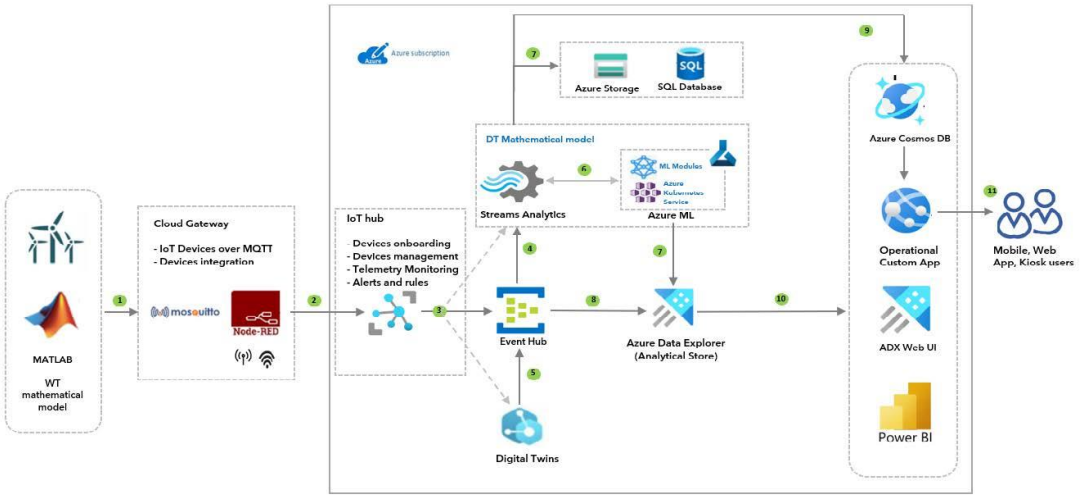

Figure 84. Digital twin and telemetry flow of a wind turbine with Azure IoT [24].

The data telemetry flow system for the wind turbine begins with sensors measuring and collecting real-time data such as wind speed, turbine speed, and mechanical power. It is transmitted to the Azure IoT Hub using the MQTT protocol. The IoT Hub forwards the data to Azure Digital Twins (ADT), where the virtual model updates in real-time, while simultaneously passing through Azure Stream Analytics for preprocessing. Azure Machine Learning analyzes the data to predict failures and identify operational trends. These insights are fed back to the turbine for optimization, and dashboards present actionable metrics for operators to monitor and make informed decisions [24].

6.2. Smart City Platform

In a FIWARE-based smart city system, data is generated by IoT sensors, like parking sensors, and user inputs, such as restaurant reviews. This data is transmitted via IoT agents (e.g., LoRaWAN), which convert the information into NGSI-compliant format for the Orion Context Broker. User actions, like submitting reviews, are directly sent to Orion using POST requests. The Orion Context Broker manages this real-time data by aggregating it into entities (e.g., parking spots) and supporting subscriptions to notify users of relevant updates. A notification system based on Orion's subscribe/publish model keeps users or businesses informed about changes, like parking availability. Historical data storage is handled by Cygnus, which logs data changes into a database, allowing

analysis of trends, such as parking patterns [19].

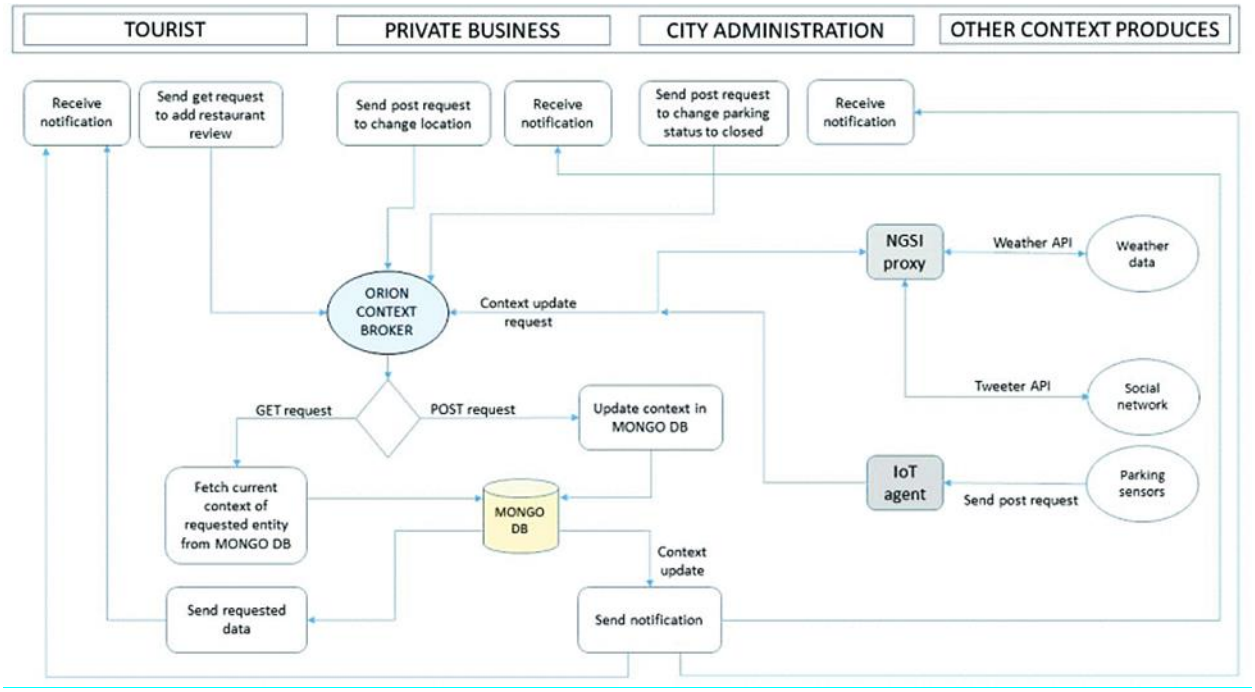

Figure 95. FIWARE based smart city platform [19].

6.3. Smart Farm Management

The FIWARE IoT platform presents advanced architecture for smart farming, integrating diverse components to improve efficiency and productivity. At its core, the Orion Context Broker consolidates real-time data from sensors, drones, and robots, offering a centralized view of critical metrics such as soil moisture and crop health for streamlined farm management. To ensure effective communication, IDAS IoT Agents connect sensors using protocols like MQTT and CoAP, translating their data into a format compatible with the Orion Context Broker [27].

Robotic systems, powered by ROS-2, manage autonomous equipment like drones and tractors through real-time Fast RTPS communication, enhancing precision in tasks and reducing reliance on manual labor. Complementing real-time data, Hadoop and Spark process historical information, analyzing trends in weather patterns and crop yields to support predictive strategies and resource planning. *Wirecloud* and *Knowage* dashboards build on these insights, offering real-time visualization tools and performance metrics to assist farmers in making informed decisions aligned with

sustainability goals. Finally, CKAN facilitates the secure sharing of agricultural data with researchers and agrotechnology companies, creating opportunities for innovation and revenue generation [27]. This architecture demonstrates how IoT technologies can transform traditional farming practices into efficient, data-driven systems.

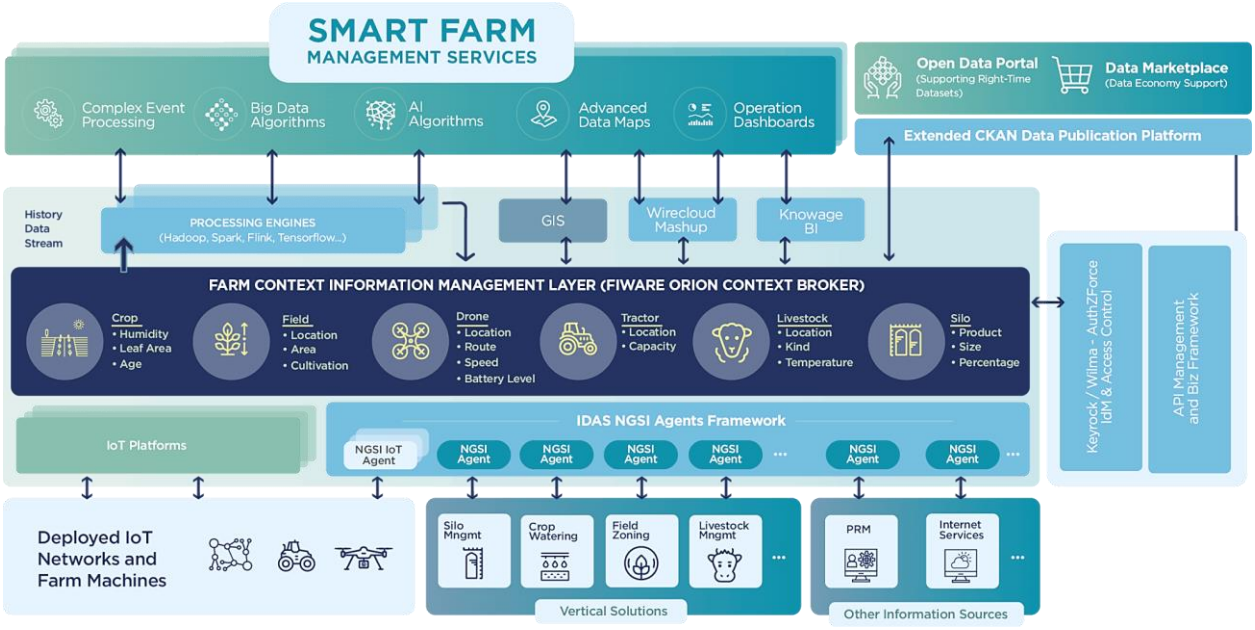

Figure 106. A Reference Architecture for Smart Farm Management Powered by FIWARE [27].

7. Challenges and Future Issues

7.1. Interoperability and Standardization

Interoperability and standardization in IoT platforms like Azure, AWS, and FIWARE face challenges from heterogeneity, vendor lock-in, resource constraints, and scalability. Proprietary technologies and diverse protocols create silos, hindering communication between devices. Vendor-specific ecosystems restrict flexibility and increase costs. Limited device resources strain complex interoperability protocols, affecting performance. As IoT systems scale, maintaining seamless communication across devices and platforms becomes more challenging, emphasizing the need for common standards to ensure efficient and reliable IoT ecosystems.

7.2. Security and Privacy Concerns

IoT systems face significant security and privacy challenges due to their distributed nature and the use of resource-constrained devices. Common vulnerabilities include weak authentication methods, insecure firmware, unencrypted communication, and lack of regular security updates (as shown in Table. 3). These flaws expose IoT devices to risks such as unauthorized access, data interception, and denial-of-service attacks. Privacy issues also arise from the vast amount of sensitive personal data IoT systems generate, requiring robust encryption, access control, and compliance with privacy regulations like GDPR to protect user information [28].

Table 3. Major security flaws in common IoT devices.

Devices	Vulnerability	Impact	Likelihood
Smart Thermostat	Weak default Password	Unauthorized access, Temperature Manipulation	High
Wearable Fitness Tracker	Insecure Bluetooth pairing	Data Interception, Privacy Breach	Medium
Smart Security Camera	Lack of Firmware Updates	Vulnerability Exploitation, unauthorized access	Low

7.3. Scalability and Network Management

Scalability is critical for IoT systems, which often grow to include thousands of devices. Effective network management ensures stable performance despite high data traffic. Challenges include managing bandwidth, device registration, and real-time communication. Techniques like dynamic network slicing and load balancing, along with protocols such as CoAP and MQTT, help optimize data flow. Cloud and edge computing frameworks further support scalability by distributing computational loads and reducing latency.

7.4. Energy Efficiency and Sustainability

IoT systems must address energy efficiency due to their reliance on battery-powered devices and large-scale deployments. Energy-efficient communication protocols like Zigbee and LoRaWAN minimize power consumption during data transmission. Adaptive duty cycling in sensors and the integration of energy harvesting technologies such as solar panels extend device lifespan. Sustainability is also enhanced by smart applications in energy grids, resource optimization, and

predictive maintenance which helps reducing environmental impacts.

7.5. Data Storage and management

Data storage and management in IoT platforms encounter significant obstacles due to the vast amount of data, rapid generation, and diverse formats of data produced by interconnected devices. Challenges include managing *data heterogeneity*, which complicates integration and interpretation, and implementing scalable storage systems to accommodate increasing data sizes. Considering the sensitive nature of IoT information, safeguarding data privacy and security is equally critical, Effective processing and timely analytics are required to get meaningful insights without delay. Though cloud and edge computing address many of these issues but they introduce constraints like latency.

7.6. Regulatory and Legal Challenges

The expansion of IoT has introduced regulatory challenges like privacy, data protection, device certification, and data sovereignty. Compliance with laws like GDPR and CCPA ensures data protection and user rights. Certification standards address device safety and reliability. Regulations like the EU Data Act emphasize localized data storage to protect privacy. In healthcare, IoT-enabled systems raise ethical concerns by balancing data usage for better care with privacy and informed consent which highlights the need for proper governance frameworks.

8. Conclusion

The evolution of IoT platforms has opened new opportunities for future development as well as some risk factors associated with them. Regardless of the risks, IoT is one of the most important aspects of technology. We cannot imagine our daily lives nowadays without IoT. We are surrounded by smart devices, everything connected to cloud, sharing trillions of bytes of data every single day. IoT has not only merged the physical gap by eliminating travel constraints but also, they have made our lives easier by eliminating the need to be present somewhere physically to be able to control a smart device.

We thereby conclude that everyone should contribute to the field of IoT to the best of their knowledge in order to improve the scope of IoT development in future.

References

1. Babun, Leonardo, et al.,"A survey on IoT platforms: Communication, security, and privacy perspectives," Computer Networks, vol. 192, 2021.
2. K. Shafique, et al., "Internet of Things (IoT) for Next-Generation Smart Systems: A Review of Current Challenges, Future Trends and Prospects for Emerging 5G-IoT Scenarios," in IEEE Access, vol. 8, pp. 23022-23040, 2020.
3. Arshi, O., Mondal, S., "Advancements in sensors and actuators technologies for smart cities: a comprehensive review," Smart Construction and Sustainable Cities, 2023.
4. Article by PCB HERO, "Using Microcontrollers in the Internet of Things (IoT) Applications", Nov 2024. URL:https://www.pcb-hero.com/blogs/lickys-column/using-microcontrollers-in-the-internet-of-things-iot-applications
5. Tutorial by Arduino, "Arduino UNO R3 - Overview of the Arduino UNO Components",2022. URL: https://docs.arduino.cc/hardware/uno-rev3/
6. K. Elgazzar, et al., "Revisiting the internet of things: New trends, opportunities and grand challenges," Frontiers in the Internet of Things, vol. 1, 2022.
7. M Jamuna, et al., "A Study of Communication Protocols for Internet of Things (IoT) Devices: Review", 3rd International Conference on Integrated Intelligent Computing Communication & Security (ICIIC 2021), pp. 262-268, 2021.
8. Quy, N.M., Ngoc, L.A., et al., "Edge Computing for Real-Time Internet of Things Applications: Future Internet Revolution," Wireless Personal Communications, 132, 1423–1452, 2023.
9. Rupanetti, Dulana, et al., " Combining Edge Computing-Assisted Internet of Things Security with Artificial Intelligence: Applications, Challenges, and Opportunities" Applied Sciences 14, no. 16: 7104, 2024.
10. M. Mansour, et al., "A. Internet of Things: A Comprehensive Overview on Protocols, Architectures, Technologies, Simulation Tools, and Future Directions. Energies, 16, 3465, 2023.
11. Amirian, H., Dalvand, K. et al., "Seamless integration of Internet of Things, miniaturization, and environmental chemical surveillance", Environmental Monitoring and Assessment, 196, 582, 2024.
12. Ye, L., Wang, et al., "Research progress on low-power artificial intelligence of things (AIoT) chip design. Sci. China Inf. Sci. 66, 200407, 2023.
13. A. Mijuskovic, et al., "Comparing Apples and Oranges in IoT Context: A Deep Dive Into Methods for Comparing IoT Platforms," in IEEE Internet of Things Journal, vol. 8, no. 3, pp. 1797-1816, 1 Feb.1, 2021
14. O. Toutsop, K. Kornegay and E. Smith, "A Comparative Analyses of Current IoT Middleware Platforms," 2021 8th International Conference on Future Internet of Things and Cloud (FiCloud), Rome, Italy, 2021, pp. 413-420.
15. T. D. Bolano, et al.,,"An overview of IoT architectures, technologies, and existing open-source projects," Internet of Things, Vo. 20, 100626, 2022.

16. S. K, A. X. K, D. Davis and N. Jayapandian, "Internet of Things and Cloud Computing Involvement Microsoft Azure Platform," 2022 International Conference on Edge Computing and Applications (ICECAA), Tamilnadu, India, 2022.
17. Guth, J. et al., "A Detailed Analysis of IoT Platform Architectures: Concepts, Similarities, and Differences," Internet of Everything, Internet of Things, Springer, 2017.
18. F. Cirillo, G. Solmaz, E. L. Berz, M. Bauer, B. Cheng and E. Kovacs, "A Standard-Based Open Source IoT Platform: FIWARE," in IEEE Internet of Things Magazine, vol. 2, no. 3, pp. 12-18, September 2019.
19. A. Sinanovic, et al., "Smart city use case development based on FIWARE technology," 2022 30th Telecommunications Forum (TELFOR), pp. 1-4, Belgrade, Serbia, 2022.
20. J. Kristan, P., et al.,"Evolving the Ecosystem: Eclipse Arrowhead integrates Eclipse IoT," NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium, pp. 1-6, Budapest, Hungary, 2022.
21. A. S. Muhammed and D. Ucuz, "Comparison of the IoT Platform Vendors, Microsoft Azure, Amazon Web Services, and Google Cloud, from Users' Perspectives," 2020 8th International Symposium on Digital Forensics and Security (ISDFS), pp. 1-4, Beirut, Lebanon, 2020.
22. R. Issa, M. S.Hamad and M. Abdel-Geliel, "Digital Twin of Wind Turbine Based on Microsoft® Azure IoT Platform," 2023 IEEE Conference on Power Electronics and Renewable Energy (CPERE), Luxor, Egypt, pp. 1-8, 2023.
23. Guoping Rong, et al., "An edge-cloud collaborative computing platform for building AIoT applications efficiently," Journal of Cloud Computing 10, 1, Dec 2021.
24. J. Koo, G. Kang and Y. -G. Kim, "Access Control Framework for Cross-Platform Interoperability in the Industrial Internet of Things," in IEEE Transactions on Industrial Informatics, 2024.
25. Patel B., et al., "Operating system support, protocol stack with key concerns and testbed facilities for IoT: A case study perspective," in Journal of King Saud University - Computer and Information Sciences, vol. 34 (8), pp. 5420-5434, 2022.
26. S. Hans, et al., "AI Integration in IoT: A Comprehensive Overview of Applications and Implications," 2023 Second IEEE International Conference on Measurement, Instrumentation, Control and Automation (ICMICA), Kurukshetra, India, 2024, pp. 1-6
27. Article by FIWARE, "SMART AGRIFOOD," June, 2023.
URL: https://www.fiware.org/community/smart-agrifood/
28. K. K. S. Gautam, et al., "Investigation of the Internet of Things (IoT) Security and Privacy Issues," 2023 5th International Conference on Inventive Research in Computing Applications (ICIRCA), pp. 1489-1494, Coimbatore, India, 2023.
29. White Paper by Matt Hatton, "Position Paper: Meeting the increasing regulatory challenge in IoT", Transforma Insight, Oct 2024.
30. Richard Rak, "Internet of Healthcare: Opportunities and Legal Challenges in Internet of Things-Enabled Telehealth Ecosystems," in 14th International Conference on Theory and Practice of Electronic Governance (ICEGOV 2021), October 06-08, Athens, Greece, 2021.