ISSN NO-2584-2706

Architecture and Solutions in IoT Middleware: A Comprehensive Review

Dr. Nitika Arora

Introduction

The Internet of Things (IoT) provides the ability for humans and computers to learn and interact from billions of things that include sensors, actuators, services, and other Internet-connected objects. The realization of IoT systems will enable seamless integration of the cyber world with our physical world and will fundamentally change and empower human interaction with the world. A key technology in the realization of IoT systems is middleware, which is usually described as a software system designed to be the intermediary between IoT devices and applications. Middleware technologies for the Internet of Things (IoT) play a crucial role in bridging the gap between the diverse hardware devices and the applications that use their data. In an IoT environment, devices often differ in terms of communication protocols, data formats, and processing capabilities. Middleware solutions for the Internet of Things (IoT) serve as a critical bridge between heterogeneous devices, networks, and applications, enabling seamless communication, data management, and service integration. Given the diversity and scale of IoT ecosystems, middleware provides essential functions such as device abstraction, interoperability, scalability, security, and real-time data processing. Various types of middleware architectures—including service-oriented, agent-based, event-driven, database-focused, cloud-based, and application-based solutions—address different operational needs, from dynamic service management to big data analytics and edge computing. Platforms like AWS IoT, FIWARE, Eclipse IoT, and Apache Kafka exemplify how middleware can support robust, flexible, and efficient IoT deployments across industries such as smart cities, healthcare, manufacturing, and agriculture. By simplifying development and operation, middleware solutions are fundamental to the growth, scalability, and intelligence of next-generation IoT applications.

Keywords: Internet of Things (IoT), WSNs, radio frequency identification (RFID), virtual machine, events, services, middleware architecture, context awareness, machine-to-machine (M2M) communication

Need for middleware in Iot

The Internet of Things (IoT) envisages a future in which digital and physical things or objects (e.g., smartphone, TVs, cars) can be connected by means of suitable information and communication technologies, to enable a range of applications and services (Ngu et al., 2016). The IoT's characteristics, including an ultra-large-scale network of things, device and network level heterogeneity, and large numbers of events generated spontaneously by these things, will make development of the diverse applications and service a very challenging task. In general, middleware can ease a development process by integrating heterogeneous computing and communications devices, and supporting interoperability within the diverse applications and services. The need for middleware in IoT arises from the inherent complexity and diversity of IoT ecosystems.

DOI: https://doi.org/10.5281/zenodo.15379472

ISSN NO-2584-2706

IoT networks often consist of numerous devices with varying hardware capabilities, operating systems, communication protocols, and data formats. Without a common platform to manage these differences, integrating and scaling IoT systems would be extremely challenging. The middleware, which is a software application, can hide the things details from the applications by communicating with the heterogeneous connected devices/things, filtering the raw captured data, and processing them before dissemination to the connected applications, and therefore easing the backend applications' development and offering multiple common services. Middleware addresses important tasks such as data aggregation, security enforcement, device management, and service discovery, allowing developers to focus on building applications rather than dealing with low-level integration issues by providing a unified communication layer, facilitating interoperability among heterogeneous devices and platforms (Razzaque et al., 2015). It also moreover, as IoT systems grow in size and complexity, middleware ensures scalability, reliability, and efficient resource utilization, making it an essential component for the successful deployment and operation of IoT solutions (Zhang et al., 2021). A major challenge faced by application developers today is finding the most appropriate IoT middleware solution in terms of the provided functionalities that should meet the application requirements and the underlying used technologies. Therefore, the existing works on IoT middleware architecture need to be analyzed to address their existing technical challenges, issues, and gaps in this domain and suggest further improvements.

Iot Middleware Architecture

In IoT middleware architecture, the Perception Layer forms the bottom-most layer and is responsible for sensing and collecting data from the physical environment. It includes various IoT devices such as sensors, RFID tags, cameras, and actuators, which detect parameters like temperature, motion, humidity, and location. Above this, the Transportation Layer ensures the secure and efficient transmission of the collected data from the perception devices to the data processing centers or cloud platforms. It uses various communication technologies like Wi-Fi, 5G, Bluetooth, ZigBee, and LPWAN to carry the data across networks. Finally, the Application Layer sits at the top and interacts directly with users or other systems. It processes and utilizes the transmitted data to deliver intelligent services and applications across different sectors, such as smart cities, healthcare, industrial automation, and smart homes. This layered structure ensures that IoT systems are organized, scalable, and able to integrate diverse devices and technologies seamlessly (Amaral et al., 2016) as shown in figure 1..

DOI: https://doi.org/10.5281/zenodo.15379472

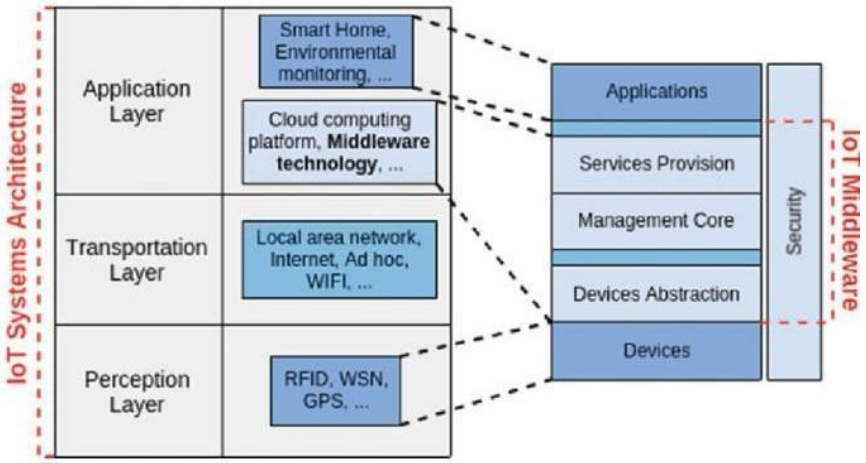

Figure 1: IoT middleware architecture

Middleware Solutions In Iot

Service oriented middleware solutions

In the Internet of Things (IoT), service-oriented middleware solutions play a crucial role in managing the complexity and heterogeneity of devices, networks, and data formats. These middleware solutions act as a bridge between IoT devices and applications by providing standardized communication, interoperability, and service management (Khaddar, 2021). They enable seamless integration of diverse systems by abstracting the underlying hardware and communication protocols, allowing developers to focus on application logic rather than device-specific details. Service-oriented middleware typically offers functionalities such as service discovery, service composition, and dynamic binding, which are essential for creating flexible and scalable IoT ecosystems. Examples include platforms like FIWARE, Kaa, and Eclipse IoT, which offer frameworks to develop, deploy, and manage IoT services efficiently. By using a service-oriented approach, middleware ensures that IoT solutions can evolve, adapt, and integrate with emerging technologies, enhancing system robustness, scalability, and ease of maintenance (Teixeira et al., 2011). Here's a quick comparison chart based on Scalability, Security, and Ease of Use for the same middleware solutions in table 1:

Middleware Platform	Scalability	Security	Ease of Use
FIWARE	**High(cloud-native, scalable microservices)**	**Good (OAuth2, secure APIs)**	**Moderate (requires understanding NGSI standards)**
Kaa IoT Platform	**Very High (designed for large deployments)**	**Very Good (encryption, secure protocols)**	**Easy (developer-friendly with good documentation)**
Eclipse IoT	**High (modular and flexible)**	**Good (depends on chosen modules)**	**Moderate (some technical expertise needed)**
ThingWorx	**Very High (enterprise-grade scalability)**	**Excellent (enterprise security standards)**	**Easy (drag-and-drop tools and visual development)**
OpenIoT	**Moderate (good for small to medium deployments)**	**Moderate (basic security, needs customization)**	**Moderate (open-source but setup can be complex)**

DOI: https://doi.org/10.5281/zenodo.15379472

Table 1: Service oriented middleware solutions

Cloud-based middleware solutions

Cloud-based middleware solutions for IoT are essential for managing the massive volume of data generated by connected devices and ensuring seamless communication between them. These middleware platforms leverage the scalability, flexibility, and computing power of the cloud to provide services such as device management, data storage, real-time analytics, security, and application integration. They enable IoT devices to connect easily to cloud servers, allowing for remote monitoring, control, and data processing (Kodakandla, 2022). Examples of popular cloud-based IoT middleware include AWS IoT, Microsoft Azure IoT Hub, Google Cloud IoT Core, and IBM Watson IoT. These platforms offer features like secure device connectivity, message brokering, rules engines for event handling, and integration with AI and machine learning services. By utilizing cloud-based middleware, organizations can quickly build and scale IoT applications without the need for extensive on-premises infrastructure, ensuring faster deployment, better resource management, and higher reliability (*Introduction to Cloud-Based Middleware in IoT*, 2021). Here's a quick comparison chart based on Scalability, Security, and Ease of Use for the same middleware solutions in table 2:

Cloud IoT Middleware Platform	Pricing	Support	Deployment Flexibility
AWS IoT	**Pay-as-you-go model (charges based on usage)**	**24/7 support available (paid plans)**	**Flexible (can be deployed globally on AWS infrastructure)**
Microsoft Azure IoT Hub	**Pay-as-you-go with free tier (limited usage)**	**24/7 support (paid plans)**	**Highly flexible (global availability and hybrid options)**
Google Cloud IoT Core	**Pay-as-you-go (pricing based on data volume and connections)**	**24/7 support (paid plans)**	**Flexible (supports hybrid and multi-cloud deployments)**
IBM Watson IoT	**Pricing based on usage (can vary)**	**24/7 support with enterprise options**	**Flexible (cloud and on-premises hybrid deployment options)**
Oracle IoT Cloud	**Pay-as-you-go (pricing based on services used)**	**24/7 support with paid plans**	**Flexible (supports hybrid, edge, and cloud deployment)**

Table 2: Cloud-based middleware solutions

DOI: https://doi.org/10.5281/zenodo.15379472

Agent Based Middleware Solutions

Agent-based middleware solutions in IoT leverage autonomous software agents to facilitate the interaction between IoT devices, applications, and users. These agents act as intermediaries that can independently make decisions, execute tasks, and manage device interactions based on predefined rules or real-time data. In agent-based middleware, each IoT device or component is often associated with an intelligent agent that can perform local processing, data filtering, and decision-making. This decentralized approach enhances the scalability and robustness of IoT systems by allowing devices to operate autonomously and respond to changes in their environment without relying on constant communication with a central server. Agent-based middleware solutions are particularly useful in scenarios where real-time processing is crucial, or where devices operate in dynamic or remote environments (Savaglio et al., 2019). Examples include the use of mobile agents in smart cities for traffic management or environmental monitoring, and autonomous agents in industrial IoT (IIoT) systems for predictive maintenance. These solutions offer advantages like reduced latency, improved efficiency, and greater fault tolerance by minimizing the need for centralized control and relying on distributed, intelligent decision-making. Here's a comparison table based on Pricing, Support, and Deployment Flexibility for some agent-based IoT middleware solutions in table 3:

Agent-BasedIoT Middleware Platform	Pricing	Support	Deployment Flexibility
JADE (Java Agent DEvelopment Framework)	Open-source (free to use)	Community-based support (via forums, documentation)	Flexible (can be deployed on various platforms like cloud, on-premises, and embedded systems)
FIPA (Foundation for Intelligent Physical Agents)	Open-source (free to use)	Community-based support (via forums, documentation)	Flexible (supports cross-platform deployment and various agent communication protocols)
CArtAgO (Cognitive Agent Theory and Applications for the Internet of Things)	Open-source (free to use)	Community-based support	Flexible (supports integration with different IoT systems and cloud environments)
Open MAS (Open Multi-Agent Systems)	Open-source (free to use)	Community-based support (limited compared to commercial options)	Highly flexible (designed for distributed environments and supports cloud and edge deployments)
AAL (Ambient Assisted Living) Agent Framework	Free for research; commercial versions available	Support through partnerships, with tailored services for commercial users	Flexible (supports deployment in healthcare, smart homes, and urban IoT environments)

Table 3: Agent-based middleware solutions

DOI: https://doi.org/10.5281/zenodo.15379472

Event Based Middleware Solutions

Event-based middleware solutions in IoT are designed to handle the asynchronous nature of data generation by IoT devices. In these systems, the communication and data flow between devices and applications are triggered by events, which are specific changes in the state or status of a device, environment, or system. When an event occurs, the middleware detects it and reacts accordingly, often by notifying relevant components or triggering certain actions. This event-driven architecture helps reduce unnecessary data transmission and processing by ensuring that actions are only taken when needed, enhancing system efficiency and reducing latency. Event-based middleware is highly beneficial in scenarios where real-time responses are required, such as in smart home automation, industrial monitoring, and security systems. These platforms typically incorporate event handlers, event queues, and message brokers, which allow IoT devices to send event notifications to other devices or cloud services. Popular examples of event-based middleware solutions in IoT include MQTT (Message Queuing Telemetry Transport), Apache Kafka, and Amazon SNS (Simple Notification Service), which are all designed to efficiently manage event-driven communication and ensure that systems remain responsive and scalable (Pramukantoro & Anwari, 2018). Here's a comparison table based on Pricing, Support, and Deployment Flexibility for some event-based IoT middleware solutions in table 4:

Event-Based IoT Middleware Platform	Pricing	Support	Deployment Flexibility
MQTT (Message Queuing Telemetry Transport)	Open-source (free to use)	Community-based support (via forums, documentation)	Highly flexible (supports cloud, on-premises, and edge deployments)
Apache Kafka	Open-source (free to use); commercial versions available	Extensive community support, paid enterprise support available	Flexible (supports distributed and cloud-based architectures, high scalability)
Amazon SNS (Simple Notification Service)	Pay-as-you-go model (based on usage)	24/7 support (paid plans)	Very flexible (fully managed, scalable cloud service)
Apache Pulsar	Open-source (free to use); commercial support available	Strong community support and paid enterprise options	Flexible (multi-cloud, hybrid, and on-premises deployments supported)
RabbitMQ	Open-source (free to use); commercial support available	Community support, paid support from vendors (e.g., Pivotal)	Flexible (supports cloud, on-premises, and hybrid deployments)

Table4: Event-based middleware solutions.

DOI: https://doi.org/10.5281/zenodo.15379472

ISSN NO-2584-2706

Database Oriented Middleware Solutions

Database middleware solutions in IoT act as an essential bridge between IoT devices and data storage systems, ensuring that the vast amount of data generated by connected devices is efficiently stored, retrieved, managed, and processed. These middleware platforms abstract the complexity of direct database interactions from IoT devices, providing standardized APIs and services for seamless communication with various types of databases, whether relational, NoSQL, time-series, or cloud-native. In IoT environments, database middleware often handles tasks like data buffering, synchronization, indexing, and querying, helping to manage real-time and historical data streams effectively. It also plays a critical role in ensuring data consistency, security, and scalability across distributed systems. Popular database middleware solutions for IoT include InfluxDB (optimized for time-series data), Apache Cassandra (for scalable NoSQL storage), and Amazon DynamoDB (cloud-based NoSQL database) ("Middleware for Internet of Things: A Survey," 1 B.C.E.). By using database middleware, IoT applications can achieve better performance, scalability, and reliability, especially in scenarios involving massive data ingestion, real-time analytics, and long-term storage for smart cities, industrial IoT, healthcare, and environmental monitoring systems. Here's a comparison table based on Pricing, Support, and Deployment Flexibility for some database middleware solutions used in IoT in table 5:

Database Middleware Platform	Pricing	Support	Deployment Flexibility
InfluxDB	**Open-source (free); Paid cloud and enterprise editions**	**Community support; Paid professional support available**	**Very flexible (can deploy on-premises, cloud, or hybrid)**
Apache Cassandra	**Open-source (free); Commercial support via vendors like DataStax**	**Strong community; Paid enterprise support available**	**Highly flexible (cloud, on-premises, multi-cloud, hybrid)**
Amazon DynamoDB	**Pay-as-you-go (based on reads/writes and storage)**	**24/7 AWS support (paid plans)**	**Flexible (fully managed by AWS, easy integration with IoT cloud services)**
TimescaleDB	**Open-source (free); Paid enterprise version**	**Community support; Paid plans for enterprise users**	**Flexible (self-hosted or cloud deployment options)**
MongoDB Atlas	**Pay-as-you-go for cloud service; Open-source MongoDB free**	**24/7 enterprise support for paid plans**	**Highly flexible (multi-cloud, hybrid, on-premises with MongoDB)**

Table 5: Database middleware solutions in IoT

DOI: https://doi.org/10.5281/zenodo.15379472

ISSN NO-2584-2706

Application-Based Middleware Solutions

Application-based middleware solutions in IoT are designed to provide a supportive layer between IoT devices and end-user applications, simplifying the development, deployment, and management of IoT services. These middleware solutions offer ready-to-use services such as data aggregation, device communication management, analytics integration, security enforcement, and user interface support, allowing developers to focus on building applications without worrying about low-level device interactions. They often include APIs, SDKs, and development tools that speed up the creation of IoT applications for various industries like smart homes, healthcare, agriculture, and industrial automation. Examples of application-based middleware in IoT include platforms like Google Firebase for IoT, Bosch IoT Suite, and Cisco Kinetic. These platforms handle critical functions such as device provisioning, event processing, real-time data visualization, and application logic management. By using application-based middleware, organizations can accelerate time-to-market for their IoT solutions, ensure better scalability and security, and easily integrate their applications with cloud services, AI, and big data tools. Here's a comparison table based on Pricing, Support, and Deployment Flexibility for some popular application-based IoT middleware solutions in table 6:

Application-Based IoT Middleware	Pricing	Support	Deployment Flexibility
Google Firebase (for IoT)	**Pay-as-you-go (free tier available)**	**24/7 support for paid plans; strong community support**	**Flexible (cloud-native, integrates easily with Google Cloud and third-party apps)**
Bosch IoT Suite	**Subscription-based (custom pricing)**	**Enterprise-level support; dedicated customer service**	**Flexible (supports cloud, hybrid, and edge deployments)**
Cisco Kinetic	**Custom pricing based on deployment**	**Full enterprise support from Cisco**	**Highly flexible (supports multi-cloud, edge, and private deployments)**
ThingSpeak (by MathWorks)	**Free basic version; paid premium options**	**Community forums; paid support for enterprises**	**Flexible (primarily cloud-based, integrates with MATLAB and IoT devices)**
Losant	**Subscription-based (tiered pricing)**	**Paid professional support; strong documentation**	**Very flexible (cloud-first, supports edge computing and hybrid deployments)**

Table 6: Application-based IoT middleware solutions

DOI: https://doi.org/10.5281/zenodo.15379472

SummaryandFutureWork

Middleware solutions in IoT play a vital role in connecting a wide variety of devices, networks, and applications by providing services such as interoperability, device management, real-time data processing, security, and scalability. Different types of middleware—such as service-oriented, event-driven, agent-based, cloud-based, database-focused, and application-based—address specific challenges in IoT systems. These platforms abstract complexity, making it easier for developers to build, manage, and scale IoT applications across industries like healthcare, smart cities, industrial automation, and agriculture. Middleware enhances system performance, improves fault tolerance, and enables seamless integration with cloud and edge computing resources.

Looking ahead, middleware solutions must evolve to support increasingly decentralized architectures like edge and fog computing, where processing happens closer to devices to reduce latency. Future middleware will also need to integrate advanced AI and machine learning capabilities for autonomous decision-making and predictive analytics. Enhancing security, ensuring energy efficiency, and enabling dynamic resource allocation will be critical focus areas. Moreover, with the rise of 6G, blockchain, and quantum computing, middleware must adapt to new communication models, ensuring ultra-reliable, real-time, and secure IoT services. Developing lightweight, adaptive, and self-healing middleware architectures will be key to supporting the next generation of massive, intelligent IoT ecosystems.

Refernces

1. Amaral, L. A., De Matos, E., Tiburski, R. T., Hessel, F., Lunardi, W. T., & Marczak, S. (2016). Middleware Technology for IoT Systems: Challenges and perspectives toward 5G. In *Modeling and optimization in science and technologies* (pp. 333–367). https://doi.org/10.1007/978-3-319-30913-2_15

2. Bandyopadhyay, D., & Sen, J. (2011). Internet of Things: Applications and Challenges in Technology and Standardization. Wireless Personal Communications, 58(1), 49–69. https://doi.org/10.1007/s11277-011-0288-5

3.Dar, K., Tahir, M., Saif, U., & Bajwa, I. S. (2019). Fog Computing-Based IoT Applications: Challenges and Solutions. Future Internet, 11(3), 61. https://doi.org/10.3390/fi11030061

4. Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions. Future Generation Computer Systems, 29(7), 1645–1660. https://doi.org/10.1016/j.future.2013.01.010

5. Introduction to Cloud-Based Middleware in IoT. (2021, April). researchgate.net. Retrieved April 27, 2025, from https://www.researchgate.net/publication/389040611_Introduction_to_Cloud-Based_Middleware_in_IoT

6. Khaddar, M. a. E. (2021). Middleware Solutions for the Internet of Things: a survey. In *IntechOpen eBooks*. https://doi.org/10.5772/intechopen.100348

7.Kodakandla, N. (2022). CLOUD BASED MIDDLEWARE FOR IOT APPLICATION USING EDGE COMPUTING AND 5G NETWORKS. In International Journal of Novel Research and Development, *International Journal of Novel Research and Development* (Vol. 7, Issue 6) [Journal-article]. https://www.ijnrd.org/papers/IJNRD2206123.pdf

8.Middleware for Internet of Things: a Survey. (1 B.C.E.). *IEEE INTERNET OF THINGSJOURNAL*.

DOI: https://doi.org/10.5281/zenodo.15379472

https://research.tees.ac.uk/ws/portalfiles/portal/5865220/621792.pdf
9.Ngu, A. H. H., Gutierrez, M., Metsis, V., Nepal, S., & Sheng, M. Z. (2016). IoT Middleware: A Survey on Issues and Enabling technologies. *IEEE Internet of ThingsJournal*,1. https://doi.org/10.1109/jiot.2016.2615180
10.Pramukantoro, E. S., & Anwari, H. (2018). An event-based middleware for syntactical interoperability in internet of things. *International Journal of Electrical and Computer Engineering (IJECE)*, *8*(5), 3784. https://doi.org/10.11591/ijece.v8i5.pp3784-3792
11.Razzaque, M. A., Milojevic-Jevric, M., Palade, A., & Clarke, S. (2015). Middleware for Internet of Things: a survey. *IEEE Internet of Things Journal*, *3*(1), 70–95. https://doi.org/10.1109/jiot.2015.2498900
12.Razzaque, M. A., Milojevic-Jevric, M., Palade, A., & Clarke, S. (2016). Middleware for Internet of Things: A Survey. IEEE Internet of Things Journal, 3(1), 70–95. https://doi.org/10.1109/JIOT.2015.2498900
13.Savaglio, C., Ganzha, M., Paprzycki, M., Bădică, C., Ivanović, M., & Fortino, G. (2019). Agent-based Internet of Things: State-of-the-art and research challenges. *Future Generation Computer Systems*, *102*, 1038–1053. https://doi.org/10.1016/j.future.2019.09.016
14.Sheng, Z., Yang, S., Yu, Y., Vasilakos, A. V., McCann, J. A., & Leung, K. K. (2013). A Survey on the IETF Protocol Suite for the Internet of Things: Standards, Challenges, and Opportunities. IEEE Wireless Communications, 20(6), 91–98. https://doi.org/10.1109/MWC.2013.6704479
15.Teixeira, T., Hachem, S., Issarny, V., & Georgantas, N. (2011). Service oriented middleware for the Internet of Things: A perspective. In *Lecture notes in computer science* (pp. 220–229). https://doi.org/10.1007/978-3-642-24755-2_21
16.Zhang, J., Ma, M., Wang, P., & Sun, X. (2021). Middleware for the Internet of Things: A survey on requirements, enabling technologies, and solutions. *Journal of Systems Architecture*, *117*, 102098. https://doi.org/10.1016/j.sysarc.2021.102098

DOI: https://doi.org/10.5281/zenodo.15379472