DOI: 10.37943/16ENNA6243

Kenzhegali Nurgaliyev
PhD candidate, Department of Information Systems
ken.nurgali@yandex.kz, orcid.org/0000-0001-9391-0924
L.N. Gumilyov Eurasian National University, Kazakhstan

Akylbek Tokhmetov
Candidate of Physical and Mathematical Sciences, Associate Professor
Department of Information Systems
attohmetov@mail.ru, orcid.org/0000-0003-0764-8574
L.N. Gumilyov Eurasian National University, Kazakhstan

Liliya Tanchenko
Master of technical sciences, Department of Information Systems
ltanchenko@mail.ru, orcid.org/0000-0002-6811-2303
L.N. Gumilyov Eurasian National University, Kazakhstan

AN ANALYSIS OF THE HETEROGENEOUS IOT DEVICE NETWORK INTERACTION IN A CYBER-PHYSICAL SYSTEM

Abstract: The article is devoted to the study of existing technologies regarding Internet of Things (IoT) device interaction in a heterogeneous network. Since each smart home appliance can be controlled by a customer who aims to find a cost-effective and easy-to-connect product for their own connected home, there are certain functional limitations for devices from distinct manufacturers that may decrease the intention to merge them all into a single network. A variety of proprietary protocols and communication standards embedded by vendors make their products unable to interact with other vendor devices if the connection standard used is not identical. Also, an IoT product design refers to its own functionality, mainly excluding the possibility of integration into other existing infrastructure. As IoT equipment emerges on the market, the complexity of its connection to a heterogeneous network corresponds to the firmware and the standard unification according to modern demands. It means that potential users might face the necessity of overcoming these issues to achieve high performance in terms of network interoperability. In general, an IoT gateway operating as a middleware might have the potential to enable a network with distinct communication models to operate without failure or data loss. This task requires the received data to be converted into the format in which the data is intended. This paper includes a comparative analysis of existing IoT device interaction standards, connection protocols, and data transfer technologies, evaluating their features for an effective adoption of the proposed network architecture, which can be used to improve the interoperability of heterogeneous IoT devices.

Keywords: internet of things; heterogeneous; network interaction; cyber-physical system; messaging protocols; data transferring standards.

Introduction

The "cyber-physical system" term (CPS) was introduced by an employee of the US National Science Foundation, Helen Gill, in 2006 to designate a distributed system where information processing occurs directly in its components [1]. A cyber-physical system consists of devices that process information and communicate with their distributed environment using actua-

Received: 16.10.2023 Accepted 10.11.2023 Published 14.11.2023

Scientific Journal of Astana IT University
ISSN (P): 2707-9031 ISSN (E): 2707-904X
VOLUME 16, NOVEMBER 2023

tors. The interaction is carried out either using other sensors or more complex mechanisms to convert energy from the environment into electricity. In other words, CPS devices interact with and respond to stimuli from the environment. Thus, CPS is an entity with two-way communication between physical processes and computing facilities.

A systematic review of CPS technologies [2, 3] highlights the following areas: wireless sensor networks, the Internet of Things (IoT), fog and cloud computing, etc., as the closest related to the subject. Most researchers indicate that the concept of "cyber-physical system" developed from the field of embedded real-time systems [4]. Such systems involve the integration of disparate devices (for instance, fire sensors and actuators) into a single computer network and ensuring their interaction through connection protocols. The increasing complexity of how computational and physical elements should interact and what their purpose is has given rise to the need for a new interdisciplinary approach.

Article analysis regarding this topic shows that among experts there is no common consensus on the mutual correspondence of CPS and IoT terms [5]: there are opinions about the partial overlap of the meaning of these two concepts [6], their equivalence [7], and the inclusion of the CPS concept in IoT [8] and reverse inclusion [9]. In general, there is a tendency towards the convergence of the CPS and IoT concepts; from a functional point of view, modern systems are considered to fit the definition of both terms.

It has prompted numerous extensive and ongoing studies on IoT-based cyber-physical systems. This is due to the rapid growth of network technologies. Currently, the main factor in delayed IoT development is the rapid increase of terminal devices based on this technology (14.3 billion devices in 2022 [10]). The growth of smart device usage, despite the numerous advantages of their use, also reveals drawbacks. Thus, there are difficulties with big data transfer as well as making optimal decisions when deploying IoT systems and ensuring the valid interaction of devices within the same network. The latter is further complex because of the competition between manufacturers engaged in this area [10].

The market is aggravated by the rapid development and modification of wireless communication technologies. As a result, today the majority of IoT products are very heterogeneous, with numerous other vendor devices that are not compatible to connect to. This circumstance might slow down the development of the entire industry. It is not helped by the lack of uniform standards, coupled with limited practical experience in sharing different devices.

The rapid development of IoT has led to the creation of institutions developing applied standards for implementation. International organizations [11, 12] and alliances of manufacturers and operators [13] are dealing with issues of standardization and its practical implementation. Nevertheless, despite the large number of interested parties, or, conversely, due to this, the efforts undertaken are mainly local, fragmented, and aimed at solving rather narrow problems rather than addressing the main issue.

Moreover, any IoT device must connect to a limited number of different objects via the Internet and other networks, so a universal and flexible modular architecture is a key element. There is currently no generally accepted architecture model for IoT devices, and the unification of the data exchange is a challenging task to achieve heterogeneous interoperability (functional compatibility). One of the challenges for creating a reference architectural model is the natural fragmentation of IoT devices. Each device depends on many and very often different parameters and requirements for performance or effective functionality. An additional challenge is the tendency for vendors to offer only their own platforms for their proprietary IoT applications.

The purpose of this article is an analytical review of existing protocols, the development of recommendations for their use, and their classification according to key criteria. To achieve

DOI: 10.37943/16ENNA6243

this goal, it is necessary to consistently identify the features of protocols, the stages of their interaction, and the key method of classification. The methodology is to determine the common place of each IoT network stage where elements might interact more effectively.

The rest of the paper is structured as follows: The IoT network topology is viewed in terms of three domains (device, getaway, and data) where heterogeneous IoT components have the potential to interact. After, each network stage analysis is represented with a brief key feature determination. Finally, a paper provides some thoughts for future work and concluding remarks.

General IoT Network Topology

The IoT concept significantly expands the capabilities of data collection, analysis, and distribution, which turn into information (knowledge for users). The Internet of Things is becoming a technology that is used to create a system consisting of interacting intelligent autonomous objects, which are complemented by sensors and actuators.

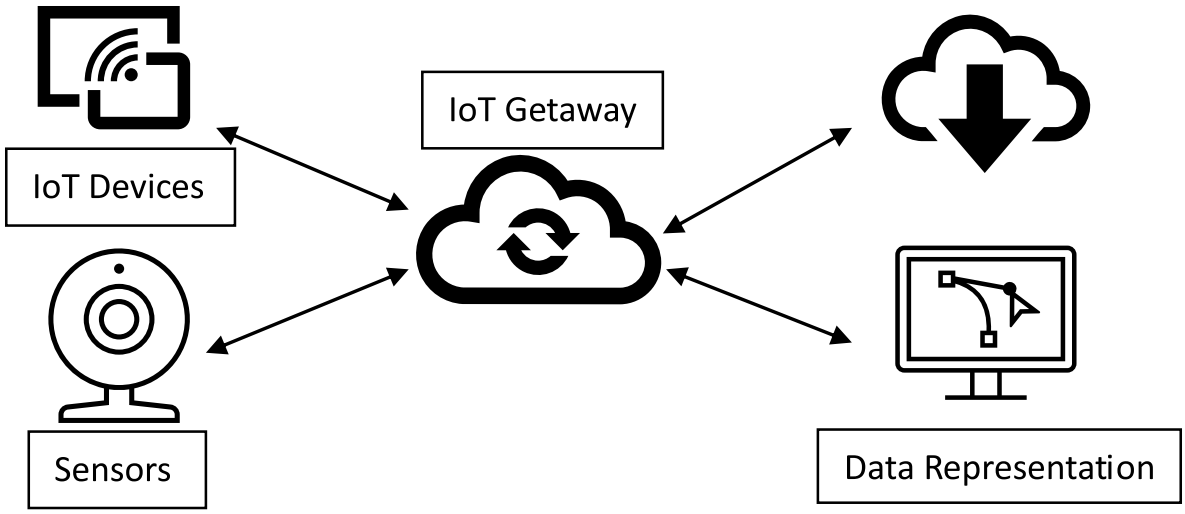

Figure 1. Heterogeneous IoT device network topology.

The architecture of the IoT network assumes the presence of the following functional domains: sensors (physical), gateways (control units), and applications for users [14]. Since the first edge level consists of actual devices, the "compatible" protocols need to ensure the interaction between sensors and with other levels (Figure 1). Standards are not suitable due to their inability to adapt to the unscripted conditions of the IoT network. For instance, the sensor, usually a miniature with a small memory, measures physical parameters in real time, most often under low-power conditions. The measurement results are processed by the sensor node and transmitted to the control unit. The amount of information generated by one sensor node is relatively small; however, most IoT services are built on the principle of processing information from many nodes, which is fundamentally different from the topologies adopted in standard documentation. Thus, we are faced with a new topology: many sensors and many receivers; in addition, the amount of traffic from a sensor node can be either very small or very large. Conventional application protocols are not designed for such use.

Also, there are several challenges, mainly related to security and device compatibility issues, at the physical level. The heterogeneity of available smart devices comes from their communication capabilities (protocols, technologies, and equipment). Therefore, this directly affects interoperability and successful adoption from users.

Scientific Journal of Astana IT University
ISSN (P): 2707-9031 ISSN (E): 2707-904X
VOLUME 16, NOVEMBER 2023

Devices Interaction (Message Protocols)

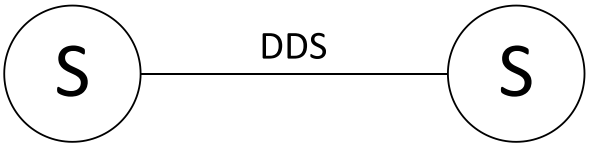

Figure 2. Interaction between sensors.

Among devices, the concept of a sensor node is usually considered. This is usually a component that can be a combination of information from several sensors (Figure 2). Therefore, many prompts can be performed inside their own network, for example, distributing information for redirection or temporary storage. The communication between sensor nodes is provided by the DDS (Data Distribution Service) protocol [15]. The message transmission itself is carried out using the request-response method. The protocol implements two basic operations: read and write. This does not remove information from the local DDS cache, and as a result, read operations can be implemented again if special parameters are specified.

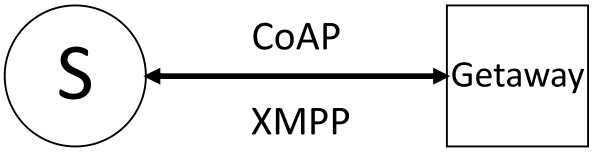

Figure 3. Interaction of a sensor with Getaway.

Regarding the communication of sensors and the gateway (Figure 3), several tasks are usually implemented, such as registering and configuring a new sensor, data transfer and its redistribution. There are two basic protocols used in this part of the network: XMPP or COAP.

XMPP (Extensible Messaging and Presence Protocol) is an extensible protocol for exchanging messages and presence data [16]. It uses device addressing using JIDs (the address of an XMPP entity, a Jabber Identifier) over a TCP connection. The messages themselves use XML text format (see Data Representation section) and are activated via request-response. Using XMPP, it is possible to connect a smart thermostat to a gateway, accessing it from a mobile phone. The advantages of this protocol are also security and scalability, which makes it suitable for creating multi-protocol solutions in a small personal Smart Home network, where the use of lighting and climate control systems dominates.

COAP (Constrained Application Protocol) is a specialized data transfer model created for devices with limited resources and low power consumption [17]. The operating principle of COAP is similar to HTTP - it also uses GET-PUT requests over a UDP connection. The gateway can use queries to control and monitor sensors. Its request will check a status flag, but data will continue to be sent (status streaming) even after the original message has been delivered. For example, this protocol is used to work with indoor temperature sensors.

Thus, the gateway can perform the functions of collecting information, organizing request queues, distributing, and storing data on demand to transfer to the server (cloud). When the load on the network increases due to the appearance of a server (cloud) in the topology, the MQTT or STOMP protocols are usually used (Figure 4).

DOI: 10.37943/16ENNA6243

Figure 4. Interaction with server (cloud).

MQTT (Message Queue Telemetry Transport) is a data exchange technology for telemetry and remote monitoring [18]. Allows smart devices to send and receive data only when a specific event occurs. This means that we have a binary protocol running on a TCP connection. The protocol uses a "request-response" and allows you to control the parameters of QoS ("quality of service"), such as characteristics of sending messages as speed, latency and data loss, thereby giving data certain priorities. Because of this, queues of requests are created that the gateway enables to classify data. This protocol is used in busy networks with a large number of devices to reduce the load on the communication channel by organizing such queues.

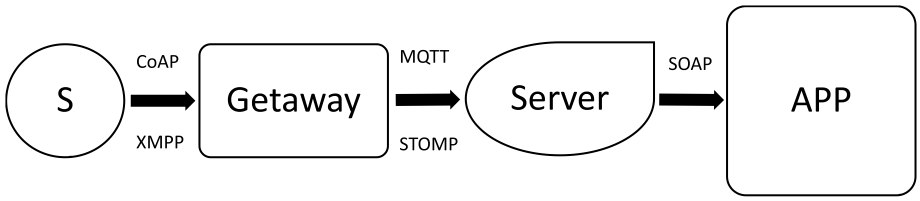

Figure 5. Interaction with an App (monitor, application).

The final section of a typical IoT network is the client stage: an application or any other tool for status representation from sensors and for data processing results. Processes related to the interaction of the IoT "client" are also performed here: setting parameters (data update configurations, activation/deactivation of sensors). Therefore, at this stage, STOMP (Streaming Text-Oriented Message Protocol – a simple data exchange standard with a "request-response" model) and SOAP (Simple Object Access Protocol, used for exchanging structured and arbitrary XML) technologies are implemented where there is a remote call of the necessary functions [19].

Table 1 depicts a comparative analysis of presented standards in terms of various communication patterns support (see Getaway Interaction section).

Scientific Journal of Astana IT University
ISSN (P): 2707-9031 ISSN (E): 2707-904X
VOLUME 16, NOVEMBER 2023

Table 1. IoT messaging protocols analysis.

Message protocol	Purpose & Key features	Request-Response	Asynchronous messaging	Discovery	Multicast Routing	Event Subscription
DDS	Only for receiving and sending data on networks with a large number of nodes. Availability of cache (history), support for basic synchronization.	Supported + publish-subscribe	Supported.	Supported.	Supported.	Not supported.
COAP	For devices with low power consumption without specific parameters. Binary protocol for networks with limited bandwidth.	Supported.	Supported.	Supported.	Not supported	Not supported.
XMPP	For network with small number of nodes. Component identification, supports device search.	Supported + publish-subscribe	Supported	Supported	Not Supported	Supported
MQTT	Used for multi-device networks procedures for recording device statuses, a request queuing mechanism. Maintains quality of service, ensures message delivery is verified.	publish-subscribe only	Supported	Not supported	Not supported	Not supported

Getaway Interaction

After conceptually creating heterogeneous network access points, the main work is to find an approach for a gateway to receive and process the actual data. Data transmitted from these networks contains a large amount of invalid information, such as processing sensor statuses. In addition, all types of sensory information are used in different formats, and interpretation will require a large amount of system resources.

To create a complex IoT hardware structure, a developer needs to take advantage of existing server technologies and, very importantly, cloud services. However, in case of sending any sensor readings from a low-power board, it is necessary to include an additional hardware link in the "sensor-server" interaction scheme (see Figure 4), ensuring reliable and cost-effective communication of the IoT device with other domains.

The main goal of an IoT gateway is to successfully implement data transformations. The gateway develops a standard format middleware protocol for various heterogeneous data during processing [14]. Any prompt can be converted to a standard format regardless of what message format is received as the corresponding data adaptation protocol script is loaded.

DOI: 10.37943/16ENNA6243

Moreover, it can ensure the integrity of the received data. The data can be analyzed according to the prior rule, extracting effective information for further processing. To build multiprotocol IoT network, a combination of wireless local, mesh and wired technologies is required; therefore, gateway should have a heterogeneous feature.

In terms of functionality, getaways can be as shown in Figure 6.

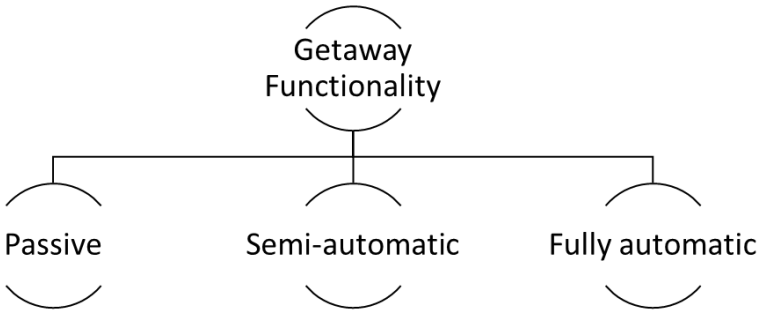

Figure 6. Functionality types of IoT getaway.

With passive gateways, new smart equipment, such as sensors, must be configured manually. For example, a user needs to remove old devices that are no longer part of the network [20]. Semi-automated gateways can immediately communicate with a new component but cannot automatically support setting all its parameters during installation [21]. Finally, fully automatic gateways allow new IoT devices to be self-configured and thus quickly solve heterogeneity problems in data transmission [22].

Before claiming the network architecture, a user should choose a gateway to achieve maximum network performance and avoid overloads and network failures for one reason or another. To do this, it is necessary to evaluate in advance the possibility of using a particular communication model for incompatible devices. As the getaway starts to develop, the IoT system risks in serious code rewriting, as it is crucial to determine how the getaway interacts with the outside elements.

REQUEST – RESPONSE (Figure 7) communication template is implemented involving a user (client), that makes requests to some service software on a server (a responder).

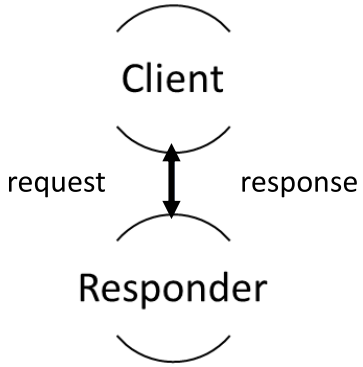

Figure 7. Request-Response interaction topology.

For instance, if any sensor is a client and an IoT gateway is a server, the sensor decides when to transmit its own readings to the server. The server, if sensor data is needed, will not be able to request it on its own. When the sensor is a server and the gateway is a client, the sensor can be polled at any time. However, in this case, anyone can connect to the gateway in terms

Scientific Journal of Astana IT University
ISSN (P): 2707-9031 ISSN (E): 2707-904X
VOLUME 16, NOVEMBER 2023

of security [23]. Thus, an interaction between the client, the server, and the system should be more complicated.

As Table 2 shows, there are some features analyzed in terms of communication patterns for an IoT gateway interaction model.

Table 2. Getaway communication models analysis.

Model	Purpose	Key Features
REQUEST – RESPONSE	Basis of service-oriented architectures, web services and REST solutions, especially if the project architecture involves master and slave entities.	Practical template used by HTTP, COAP, XMPP (see Section Device Interaction) When both entities request data from each other, it is not possible to create bidirectional data exchange
EVENT SUBSCRIPTION	Client notification in case of predefined event occurs [24]	Used by HTTP, COAP, XMPP. There is no need to constantly poll the server. Data is transferred not upon request – decrease the number of prompts.
ASYNCHRONOUS MESSAGING	Gives ability to send messages between peer systems located at the same level of hierarchy [25]	Used by XMPP. Involves bidirectional messaging.
MULTICAST ROUTING	Message delivered through an intermediate (getaway), where it is forwarded to multiple recipients [26].	Network load reduction as the single message is delivered. Data transmission with multicasting is more difficult to protect as after the intrusion there is a possibility to bypass restrictions and change to another communication model. Significant proportion of data transmitted in this way is not used by recipients.
DISCOVERY	Identification data of new devices is compared with the data of the equipped nodes in the network [27]. Used to get initial identification parameters of new devices.	Host device accesses their factory IDs as well as other nodes in IoT network. XMPP supports this pattern.

Finally, reviewing these patterns and their open specifications, it is possible to claim that the compatibility of IoT devices is reachable with existing technical models. Likewise, by using open, standardized, interchangeable components, the need to build expensive infrastructure can be avoided.

In general, a multi-protocol gateway is a technology that enables data transfer between networks with different communication models. The task of interoperability requires converting the received data into the standard format for the device for which the data is intended. In a typical scenario, a gateway can be a device that operates over various device protocols and

DOI: 10.37943/16ENNA6243

data formats. At the same time, the gateway becomes an ideal tool for creating a management interface for the entire smart home. During the exchanging of messages with network nodes (sensors, devices), it can support and collect data about them [28].

The authors [29] propose another gateway approach based on the IoTivity platform (Linux). This gateway is focused on providing interoperability between devices with non-IP-based communication capabilities. One of the advantages of this gateway is that the network immediately recognizes the new component after establishing communication with the gateway. In [30], the authors presented a use case for a multi-protocol gateway where a weather station with two wireless nodes transmits environmental data to the gateway. The gateway, acting as the central element of the system, then delivers environmental data to the cloud server. One of the main positive features is that it allows you to connect new wireless network nodes and transfer data to any data analytics service hosted in the cloud. This provides flexibility in data storage and visualization.

These multi-protocol gateways have one thing in common: sensor measurement results are processed by the gateway and transmitted to the server (cloud). The amount of information is relatively small; most sensors are quite primitive because they constantly transmit data only about the controlled parameter; however, sensors are built on the principle of endless data generation, which means the gateway needs to process this information from a huge number of such devices on the network. Some studies estimate that solving the interoperability problem requires efforts in the elimination of so-called closed ecosystems [31], while data collected and transmitted through the interaction might operate seamlessly for user benefits [32].

Moreover, research has been carried out on the development of the architecture of unified smart home gateways [33, 34], where most use an architecture based on the OSGi standard (Open Services Gateway Initiative) - a specification for Java applications with a service for installing dynamic modules without stopping and restarting the entire process [35]. The results of these studies were able to ensure that different protocols work together in a home network. For example, [33] proposed an OSGi architecture for discovering and installing new devices using barcodes and smartphones. The same approach to creating a single gateway is given in [34], where the essence is to use a set-top box or smartphone as a server hub for unified interaction via Wi-Fi with other “smart” devices. Despite this, the researchers conclude that smartphones are not always located in a fixed location in the home and cannot be the main gateway of a smart home system.

Another study [36] proposed an approach where an “openHAB” (a free home automation solution) [37] ran through a single gateway on top of other wireless interfaces, but to add a new device type, users must manually write configuration files. This task requires programming skills as well as deep technical knowledge, which is difficult for most regular clients. However, its use does not require device manufacturers to make any changes to their products; all changes are made on the server.

Using the gateway, it can implement various options for processing the messages from sensors. Even quite complex operations might create a load that will slow down the entire system. As a result, the gateway can be considered as solution for preparing data from low-power devices before sending it to the cloud or server. As getaway delivers readings, there are other approaches that can help protect data, ensure high transfer rates, and eliminate latency. The raw flow of information from the sensor to the server may seem like a simple and convenient solution, but grouping, batching information, or even collecting disparate indicators into packages and sending them to the server in the form of a single file can achieve high-quality data transfer.

In order to organize findings, Table 3 shows main proposed features of getaway after the literature review.

Scientific Journal of Astana IT University
ISSN (P): 2707-9031 ISSN (E): 2707-904X
VOLUME 16, NOVEMBER 2023

Table 3. Getaway purpose in the heterogeneous IoT network.

Getaway Purpose	Description
data aggregator	To be able to gather data coming from other devices. Small devices can handle many tasks on their own, but if they are overloaded, limitations in memory and processing power can slow down the entire network, bringing it to a near-inoperable state. Integrating devices using a gateway makes it possible to transmit data over long distances, which solves spatial location problems.
mini PC	To be equipped with single-core processors with RAM to be able to respond as quickly as possible to control inputs and changes in sensor indicators.
any network solution	To be able to connect to specialized networks with existing equipment, as it may need to disconnect from one network to connect to another without interrupting communication.

Data Interpretation

To realize unification among different message formats and to make it easier to convert them, JSON and XML are «easy» data exchange formats to meet these requirements. These are open data formats when the message itself contains field identifiers that depict the context and size of the message and determine the behavior of algorithms that read the message and provide access to its contents [38]. It allows for the creation of IoT systems that are adaptive to changes in prompt content and do not require actual support.

The XML (Extensible Markup Language) format uses a set of elements (tags) and contextual data fields (text) [39]. Tags allow to interpret the meanings of data fields and process them. This format is based on the Unicode encoding (UTF-8, UTF-16) and is easily adjusted to any specific needs of a message to be interpreted as a document.

This format may contain databases or certain application settings. The use of an XML data format in IoT serves the purpose of effective information exchange. Thus, it is convenient to exchange the required data between different IoT devices because the XML markup language is used by the owners of various operating systems.

To work correctly with XML, element names are first selected, and then the corresponding DTD (Document Type Definition) description or schema is determined based on them. There are entities which can be pieces of text or special characters inside the XML message (Figure 8). Using entities helps avoid repeating the same phrase or information during the next prompt.

DOI: 10.37943/16ENNA6243


```
<profileInformation xsi:type="deviceProfile"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <uri>
        localhost/LogicalResource/resources
    </uri>
    <name>Termometer 1</name>
    <location>
        <name>level 1 - stairs 2</name>
    </location>
    <purpose>Temperature Sensor</purpose>
    <keywords>
        <keyword>Temperature/keyword>
        <keyword>Degrees</keyword>
    </keywords>
    <model>
        <deviceClass>Termometer</deviceClass>
    </model>
    <information>
        <status>ONLINE</status>
    </information>
</profileInformation>
```

Figure 8. XML format example.

JSON (JavaScript Object Notation) is a message format based on JavaScript language syntax and offers lightweight data exchange for users and IoT devices. It can be used with any programming language, as there is ready-made code for creating and processing data in JSON format [40]. Using simple syntax, it allows to represent any data type, from a single number to strings, arrays, and objects, in plain text or by creating complex data structures (Figure 9).

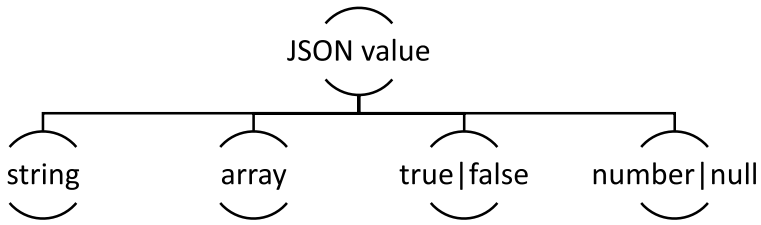

Figure 9. JSON object list.

The object (Figure 10) begins with "{" (opening brackets) and ends with "}" (closing curly brackets). Each name is followed by a ":" (colon), and key/value pairs are separated by a "," (comma).

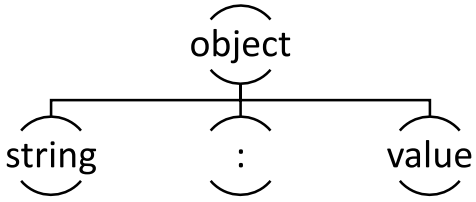

Figure 10. JSON object structure.

Scientific Journal of Astana IT University
ISSN (P): 2707-9031 ISSN (E): 2707-904X
VOLUME 16, NOVEMBER 2023

The message format between two incompatible devices can be specified by JSON package. Each message is a separate object. An object is an unordered collection of key-value pairs. The key can only be a string, and the value can be any format. Typically, JSON allows data to be exchanged without any reloading or delay and makes requests by passing GET or POST requests.

Table 4. Data Format analysis.

Data format	Advantages	Disadvantages
XML	It is convenient for the required data to be exchanged between heterogeneous platforms because the XML markup language is used in various OS. Easy to adapt to any specific needs as it is based on Unicode encoding. Attributes to elements allow additional parameters to be stored within the same message.	XML syntax is redundant. The XML-document size is significantly large. The cost of storing, processing and transmitting data increases. Modeling ambiguity. There is no generally accepted methodology for modeling data in XML. There is no data type syntax.
JSON	Universal data structure: any modern programming language (PL) can support this format as there are functions and libraries for reading and creating JSON. Syntax is tight. Allows to present any data from a single digit to strings, arrays or objects.	Not able to report the encoding format on a prompt. Security issues. Server-Client interaction is set up independently as if an additional function is called, arbitrary code is created, which in practice can already be a security risk.

Evaluation

As was mentioned above, the getaway implements the functionalities of a mediator for heterogenous cyber-physical components. It means that a possible architecture needs to be adapted to the constraints of such an IoT system. Firstly, it can be stated that all communication between incompatible devices can be based on the MQTT standard. It operates in a publish-subscribe model, meaning that a sender device does not send prompts directly to a recipient component. Instead, all messages are published, and the receiver subscribes to forthcoming messages. Thus, it hard to penetrate the code during the operation as it contributes to the security reliability of a possible IoT system.

Secondly, considering concepts of distributed computing, light-weight structured JSON avoids overhead in messages and can be encoded without special requirements by any heterogeneous component of an IoT system. Lastly, asynchronous data transmission is used to allocate resources on devices, to balance all operations and to call functions on time.

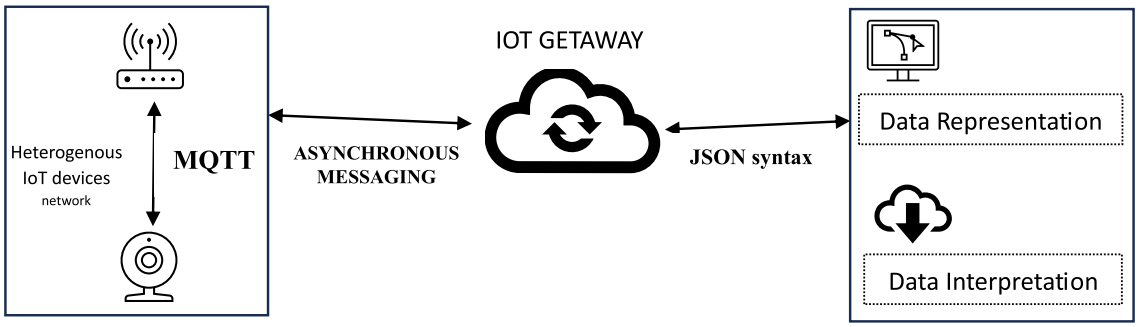

Figure 11. Proposed heterogeneous network.

DOI: 10.37943/16ENNA6243

Figure 11 depicts an example of a heterogeneous network where a getaway-based distributed architecture is highlighted. On-time programs and firmware updates, access policies and security regulations are all part of IoT device management [14]. The sensor node must get all data in a readable format. So that, the getaway requests information about the locations of files or commands to upgrade the node. Moreover, there is the universal data protocol (MQTT) to send commands to any device as a payload. This type of data can contain configuration settings that enable the device to follow security procedures (JSON objects). Security requirements are recommended to be the focus of future studies.

The given concept is designed to evaluate configuration methods from initially incompatible elements to their successful control in the IoT network. As Castellanos et al. [30] stated, interoperability can be described by the following operation management requirements (Table 5).

Table 5. IoT interoperability requirements.

Property	Feature	Importance rate
Scalability	The system can be scaled both horizontally and vertically as a user can add any device even not compatible with the current network infrastructure.	from 0 to 9
Traceability	A user can receive device real-time status, last response information.	0 - 9
Nonrepeating | Adaptability	Unable to initiate identical commands. Able to send any-sized messages.	0 - 9
Security | Safe	Getaway allows to transmit settings, an appropriate response to the requesting device without intrusions as it sends encoded data. Sensor nodes are able to deliver requests despite network or system failures as the getaway receives data using protocols.	0 - 9

To evaluate the proposed network structure, the mathematical model of the interoperability (I) of the heterogeneous IoT devices is represented as following:

$$I = S_n \times R_n \tag{1}$$

where S_n – a set of n-number of sensors or actuators and R_n – quantitative representation (importance rate) of interoperability parameters (r_n) brought to a single scale.

In other words, S_x can be described as a set of n-devices in the form of $n \times n$ matrix. Regarding R_n, this is a set of elements of size $n \times r_n$ which is matrix of interoperability parameters for evaluated interaction of each device with another. Then the combination of sets (1) characterizes the interoperability of interaction between devices.

The connection between devices and gateways can be described as a set S_g of size $n \times m$, where m – is a number of gateways used in the network. Let us define a set R_g be a size of $n \times r_g$, the elements of which are the quantitative interoperability parameters of each device when interacting with the gateway, brought to a single scale. Then (2) represents the interoperability at the connection points of the gateways:

$$S = S_g \times R_g \tag{2}$$

where S_g is a set of size $m \times r_g$.

Scientific Journal of Astana IT University
ISSN (P): 2707-9031 ISSN (E): 2707-904X
VOLUME 16, NOVEMBER 2023

Next, the relative evaluation in terms of the interoperability can be described as following:

$$A_I = \frac{\sum_y^n i_y + \sum_y^n s_y}{I_{max} + S_{max}} \quad (3)$$

where A_I – is assessment rate of device compatibility as I corresponds to device self-interaction (see Figure 1). I_{max} and S_{max} goes to maximum possible values of device-device and device-gateway interaction, correspondingly. Also, i_y – are elements of I, and S_y – are elements of S.

Due to the fact that some parameters change over time, a time-dependent model is shown as:

$$A_I = \frac{\sum_y^n i_y\,(t) + \sum_y^n s_y\,(t)}{I_{max} + S_{max}} \quad (4)$$

This model is useful to design an IoT environment and determine the current interoperability features for all three stages of interaction (see Figure 1). Working with all three models for relative assessments, it is possible to make a further quantitative and qualitative assessment of the state of the entire system. Moreover, the network entity edition as well as compatibility requirements can be accomplished by adding, changing, or deleting rows and columns in the corresponding sets.

Conclusion

Unification of data formats and message protocols in a Smart Environment will ensure the invariance of IoT products, which will eliminate dependence on the vendor and help reduce the cost of implementation, use and development for common people. The rapid development and modification of wireless communication technologies is affecting the IoT market. It is very heterogeneous, filled with a large number of devices that are not always able to interact with each other. This fact slows down the whole development of the industry. The lack of uniform standards, coupled with limited practical experience in sharing heterogeneous devices. Existing requirements for the data format and message protocols for devices have been presented and analyzed. It gains in the process of a reasonable choice of suitable protocols and developing a rational format for transmitted data.

The given features can be used in the development of IoT systems by expanding the used functionality of the devices, using various encoding and transferring principles for obtaining information and managing a variety of physical and software processes in the interests of local users.

It is important to note that security features are not the focus of this study and are recommended for future studies. Regarding the proposed network, it is obvious that it is crucial to choose a gateway to avoid overloads and predict network failure scenarios. To do this, it is necessary to assess in advance the possibility of using a particular pattern, protocol, or message format to increase the degree of system stability, as well as gateway technical properties as its RAM and processor.

As was mentioned above, generally, in a short time, it is possible to present a heterogeneous, efficient IoT network with maximum performance for several specific requirements. The development of such requirements will contribute to the future development of this topic, and, accordingly, will make life easier for IoT product users and provide new scientific issues for Smart Home researchers.

DOI: 10.37943/16ENNA6243

References

[1] Tao, F., Qi, Q., Wang, L., & Nee, A.Y.C. (2019). Digital twins and cyber–physical systems toward smart manufacturing and industry 4.0: Correlation and comparison. *Engineering*, *5*(4), 653-661.
[2] Akbarzadeh, A., & Katsikas, S. (2020). Identifying critical components in large scale cyber physical systems. In *Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops*, 230-236.
[3] Addeen, H.H., Xiao, Y., Li, J., & Guizani, M. (2021). A survey of cyber-physical attacks and detection methods in smart water distribution systems. *IEEE Access*, 9, 99905-99921.
[4] Chui, K. T., Gupta, B. B., Liu, J., Arya, V., Nedjah, N., Almomani, A., & Chaurasia, P. (2023). A Survey of Internet of Things and Cyber-Physical Systems: Standards, Algorithms, Applications, Security, Challenges, and Future Directions. *Information*, 14(7), 388.
[5] Pivoto, D. G., de Almeida, L. F., da Rosa Righi, R., Rodrigues, J. J., Lugli, A. B., & Alberti, A. M. (2021). Cyber-physical systems architectures for industrial internet of things applications in Industry 4.0: A literature review. *Journal of manufacturing systems*, 58, 176-192.
[6] Franco, J., Aris, A., Canberk, B., & Uluagac, A. S. (2021). A survey of honeypots and honeynets for internet of things, industrial internet of things, and cyber-physical systems. *IEEE Communications Surveys & Tutorials*, 23(4), 2351-2383.
[7] Mohanta, B.K., Dehury, M.K., Al Sukhni, B., & Mohapatra, N. (2022). Cyber physical system: Security challenges in internet of things system. *2022 Sixth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, 117-122.
[8] Yaacoub, J.P.A., Salman, O., Noura, H. N., Kaaniche, N., Chehab, A., & Malli, M. (2020). Cyber-physical systems security: Limitations, issues and future trends. *Microprocessors and microsystems*, 77, 103201.
[9] Yeboah-Ofori, A., Abdulai, J., & Katsriku, F. (2019). Cybercrime and risks for cyber physical systems. *International Journal of Cyber-Security and Digital Forensics (IJCSDF)*, 8(1), 43-57.
[10] Sinha, S. (2023). State of IoT 2023: Number of connected IoT devices growing 16% to 16.7 billion globally. *IoT Analytics*. https://iot-analytics.com/number-connected-iot-devices/
[11] 2413-2019–IEEE Standard for an Architectural Framework for the Internet of Things (IoT). (2020). *IEEE Standard*. https://ieeexplore.ieee.org/document/9032420
[12] *TOSCA Version 2.0*. Edited by Chris Lauwers and Calin Curescu. (2023). https://docs.oasis-open.org/tosca/TOSCA/v2.0/TOSCA-v2.0.html
[13] International Organization for Standardization. (2018). *Cards and security devices for personal identification* (ISO 14443-1:2018). https://www.iso.org/standard/73596.html
[14] Beniwal, G., & Singhrova, A. (2022). A systematic literature review on IoT gateways. *Journal of King Saud University-Computer and Information Sciences*, 34(10), 9541-9563.
[15] Du, J., Gao, C., & Feng, T. (2023). Formal Safety Assessment and Improvement of DDS Protocol for Industrial Data Distribution Service. *Future Internet*, 15(1), 24. https://doi.org/10.3390/fi15010024
[16] Malik, M.I, McAteer, I.N., Hannay, P., Syed, N.F., & Zubair, B. (2018). XMPP architecture and security challenges in an IoT ecosystem. *Proceedings of the 16th Australian Information Security Management Conference*, 62-73.
[17] Coetzee, L., Oosthuizen, D., & Mkhize, B. (2018). An analysis of CoAP as transport in an Internet of Things environment. *2018 IST-Africa Week Conference (IST-Africa)*, 1-7.
[18] Mishra, B., & Kertesz, A. (2020). The use of MQTT in M2M and IoT systems: A survey. *IEEE Access*, 8, 201071-201086.
[19] Elsadek, W.F., & Mikhail, M.N. (2018). SOAP: SDN overlay across providers for IoT cognition services. *2018 International Conference on Innovative Trends in Computer Engineering (ITCE)*, 63-70.
[20] Bansal, S., & Kumar, D. (2020). IoT ecosystem: A survey on devices, gateways, operating systems, middleware and communication. *International Journal of Wireless Information Networks*, 27, 340-364.
[21] Diyan, M., Nathali Silva, B., Han, J., Cao, Z., & Han, K. (2022). Intelligent Internet of Things gateway supporting heterogeneous energy data management and processing. *Transactions on Emerging Telecommunications Technologies*, 33(2), e3919.

Scientific Journal of Astana IT University
ISSN (P): 2707-9031 ISSN (E): 2707-904X
VOLUME 16, NOVEMBER 2023

[22] Ramírez, P. L. G., Taha, M., Lloret, J., & Tomás, J. (2019). An intelligent algorithm for resource sharing and self-management of wireless-IoT-gateway. *IEEE Access, 8,* 3159-3170.
[23] Aloul, F., Zualkernan, I., Shapsough, S., & Towheed, M. (2020). A monitoring and control gateway for iot edge devices in smart home. In *2020 International Conference on Information Networking (ICOIN)*, 696-701.
[24] Lazidis, A., Tsakos, K., & Petrakis, E.G. (2022). Publish–Subscribe approaches for the IoT and the cloud: Functional and performance evaluation of open-source systems. *Internet of Things, 19,* 100538.
[25] Arif, N.H., & Surantha, N. (2020). IoT Cloud Platform Based on Asynchronous Processing for Reliable Multi-user Health Monitoring. In *Complex, Intelligent, and Software Intensive Systems: Proceedings of the 13th International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS-2019)*, 317-330.
[26] Gadekar, P.R., Verma, A.R., & Dhotre, V.A. (2020). Multicast routing protocols for Internet of Things (IoT) applications. *Techno-Societal 2018: Proceedings of the 2nd International Conference on Advanced Technologies for Societal Applications-Volume 2*, 99-106.
[27] Khalil, K., Elgazzar, K., Seliem, M., & Bayoumi, M. (2020). Resource discovery techniques in the internet of things: a review. *Internet of Things,* 12, 100293.
[28] Bharti, M., Kumar, R., Saxena, S., & Jindal, H. (2020). Optimal resource selection framework for Internet-of-Things. *Computers & Electrical Engineering*, 86, 106693.
[29] Kang, B., & Choo, H. (2018). An experimental study of a reliable IoT gateway. *ICT Express*, 4(3), 130-133.
[30] Castellanos, W., Macias, J., Pinilla, H., & Alvarado, J. D. (2021). Internet of things: a multiprotocol gateway as solution of the interoperability problem. *arXiv preprint arXiv:2108.00098*.
[31] Koolen, C. (2023). Interoperability in IoT Ecosystems. *SSRN*, 4474625.
[32] Vila, M., Sancho, M. R., Teniente, E., & Vilajosana, X. (2023). Critical infrastructure awareness based on IoT context data. *Internet of Things*, 23, 100855.
[33] Roy, S. K., Misra, S., & Raghuwanshi, N. S. (2019). SensPnP: Seamless integration of heterogeneous sensors with IoT devices. *IEEE Transactions on Consumer Electronics*, 65(2), 205-214.
[34] Gavrila, C., Popescu, V., Fadda, M., Anedda, M., & Murroni, M. (2020). On the suitability of HbbTV for unified smart home experience. *IEEE Transactions on Broadcasting*, 67(1), 253-262.
[35] Timalsina, U., & Wang, A. (2019). Incentivizing Services Sharing in IoT with OSGi and HashGraph. *2nd International Conference on Data Intelligence and Security (ICDIS)*, 48-52.
[36] Tsakalidis, S., Tsoulos, G., Kontaxis, D., & Athanasiadou, G. (2023). Design and Implementation of a Versatile openHab IoT Testbed with a Variety of Wireless Interfaces and Sensors. *Telecom*, 4(3), 597–610.
[37] Gunge, V.S., & Yalagi, P.S. (2016). Smart home automation: a literature review. *International Journal of Computer Applications,* 975(8887-8891).
[38] Sun, P. (2018, March). Multi-Mode Iot Gateway Design and Implemention. *2018 International Conference on Mechanical, Electronic, Control and Automation Engineering*.
[39] Azzedin, F., Mohammed, S., Ghaleb, M., Yazdani, J., & Ahmed, A. (2020). Systematic partitioning and labeling XML subtrees for efficient processing of XML queries in IoT environments. *IEEE Access*, 8, 61817-61833.
[40] Andročec, D., Tomaš, B., & Kišasondi, T. (2017). Interoperability and lightweight security for simple IoT devices. *40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 1285-1291.