DOI: https://doi.org/10.15276/aait.08.2025.12
UDC 004.451.4

A systematic approach to selecting architectural patterns for IoT development

Danylo K. Chumachenko[1)]
ORCID: https://orcid.org/0009-0000-1477-534X; chumachdk@gmail.com
Vira V. Liubchenko[1)]
ORCID: https://orcid.org/0000-0002-4611-7832; lvv@op.edu.ua. Scopus Author ID: 56667638800
[1)] Odesa Polytechnic National University, 1 Shevchenko Ave. Odesa, 65044, Ukraine

ABSTRACT

The increasing complexity and scale of Internet of Things (IoT) systems, especially within industrial environments, pose significant challenges in system design, including issues of security, interoperability, scalability, and efficient resource utilization. With a wide array of architectural patterns available to address these challenges, developers often struggle to select the most suitable solutions. This paper presents a systematic methodology for evaluating and choosing the best combinations of architectural design patterns tailored for various IoT deployment scenarios. The approach begins by analyzing existing IoT design patterns and modeling their key operational characteristics. A structured template is used to describe each pattern, facilitating consistency and comparability. These descriptions are evaluated using a quality model comprising criteria such as reliability, safety, usability, responsiveness, adaptability, durability, interoperability, and security. A weighted-sum model, with adjustable criterion weights, transforms qualitative assessments into quantitative aggregated scores. This enables objective ranking of patterns and supports defensible architectural decision-making. The methodology is validated through multiple case studies, including general-purpose IoT systems (e.g., smart homes) and Industry 4.0 environments. In each case, patterns are selected based on system-specific priorities. Notably, high-performing patterns such as Cloud-on-the-Loop, Closed-Loop Control, and Role-Based Access Control align well with known best practices and demonstrate the method's practical applicability. Sensitivity analysis further confirms the approach's adaptability, illustrating how changes in evaluation weights significantly influence the resulting pattern rankings. This systematic methodology improves the reproducibility, transparency, and flexibility of IoT architecture design processes. It empowers developers to tailor architectural solutions to specific domain needs while maintaining alignment with industry standards. Future research will explore extending the methodology to emerging IoT sectors, constructing specialized pattern catalogs, and integrating the selection framework into automated design tools to further streamline the development of IoT systems.

Keywords: Internet of Things; architectural design patterns; pattern selection methodology; quality evaluation model; system architecture

For citation: **Chumachenko D. K., Liubchenko V. V. "A systematic approach to selecting architectural patterns for IoT development".** ***Applied Aspects of Information Technology.*** **2025; Vol.8 No.2: 178–190. DOI: https://doi.org/10.15276/aait.08.2025.12**

INTRODUCTION

The Internet of Things (IoT) has fundamentally transformed device interaction by enabling networks of interconnected objects that communicate and exchange data [1]. As IoT systems continue to grow in scale and complexity, particularly within industrial environments, developers are increasingly confronted with a range of critical challenges [2]. These include ensuring system security, achieving interoperability among heterogeneous devices, managing scalability, and optimizing resource utilization. The diversity of devices and communication protocols inherent in IoT ecosystems further exacerbates these difficulties.

The design of IoT systems frequently involves the application of architectural design patterns, each intended to address specific challenges or constraints. However, the abundance of available patterns and the absence of a standardized framework for their selection often complicates the design process. This extended development timelines, and increased vulnerability to system failures or security threats.

Design patterns represent established, reusable solutions to common architectural problems [3]. By adopting these patterns, developers can construct more reliable, efficient, and secure IoT systems while reducing both complexity and development time. Patterns encapsulate expert knowledge and industry best practices, offering modular, adaptable blueprints that promote maintainability and scalability. Moreover, their consistent use fosters interoperability by establishing a shared vocabulary and structure that enhances communication among development teams.

To address the challenges associated with IoT system design and the selection of appropriate architectural solutions, this paper presents an analysis of existing IoT design patterns. It introduces a systematic methodology for selecting optimal combinations of patterns. The proposed approach is validated by aligning its recommendations with established practices in the context of Industrial IoT systems, demonstrating its practical relevance and applicability.

1. ANALYSIS OF LITERARY DATA

The IoT integrates a diverse set of devices, including sensors, controllers, smart appliances, and actuators, into interconnected networks that facilitate the collection, exchange, and processing of data. The design of IoT systems presents considerable challenges due to their large-scale, dynamic nature and the inherent security concerns associated with resource-constrained devices. A proven strategy for addressing these challenges is the application of architectural design patterns, which provide reusable, well-tested solutions to recurring problems in IoT development [4].

Several categories of design patterns are particularly relevant for IoT systems. Security-oriented patterns [5] are essential, as IoT networks frequently handle sensitive data and interface with physical infrastructure. Without adequate safeguards, these systems are vulnerable to cyber threats that may result in data breaches, service disruptions, or bodily harm.

Authentication and authorization patterns [6], [8] ensure that only verified users and devices gain access to the system. In distributed architectures with multiple endpoints, these patterns significantly enhance security and reduce the likelihood of unauthorized access.

Client-server and peer-to-peer patterns [9], [10], [11] provide foundational communication models that support efficient data exchange and distributed processing. Their use is crucial for achieving scalable and reliable interactions among heterogeneous devices.

Self-adaptive system patterns [12], [13], [14] are becoming increasingly relevant in modern IoT architectures, enabling systems to dynamically adjust to changes in the environment or their internal state. These patterns contribute to greater system resilience, fault tolerance, and operational flexibility.

Edge and fog-level patterns [15] optimize performance by relocating computation closer to data sources. Edge computing reduces latency and conserves bandwidth by processing data locally, while fog computing introduces intermediate processing layers that enable scalable, real-time analytics between edge devices and the cloud.

Industrial IoT (IIoT) systems [16], [17], [18], [19] require specialized architectural approaches to ensure high efficiency, operational safety, and system reliability in production settings. Similarly, healthcare IoT patterns [20], [21], [22] support real-time monitoring and diagnostics while ensuring secure and reliable communication across heterogeneous medical devices.

Additionally, frameworks such as the Statechart Template Library (STL4IoT) [23], [24] provide reusable components for modeling sensors, actuators, and communication flows, thus accelerating development and testing processes.

While numerous IoT design patterns have been proposed, existing studies tend to focus on specific domains and lack a comprehensive, structured framework for evaluating and selecting them. Consequently, there is a need for a systematic methodology that generalizes the selection process and supports consistent decision-making across diverse IoT implementation scenarios. This paper addresses this gap by proposing a universal, criterion-based approach for selecting optimal architectural design patterns in IoT system development.

2. THE PURPOSE AND OBJECTIVES OF THE RESEARCH

This research aims to propose a systematic methodology for evaluating and selecting architectural design patterns suitable for IoT systems. The primary objective is to identify optimal combinations of patterns that address the specific requirements of IoT applications. To this end, the study conducts an analysis of existing IoT design patterns, defines a formal set of evaluation criteria, and introduces a structured approach for pattern selection. The applicability of the proposed methodology is validated through case studies in industrial IoT settings, providing clear and actionable guidelines to support informed and defensible architectural decisions across various IoT domains.

3. COMPARATIVE MODEL FOR IOT DESIGN PATTERNS

A broad spectrum of patterns offers developers tremendous flexibility in tailoring solutions to meet a project's specific needs. At the same time, research on approaches similar in scope indicates that a core set of patterns can cover the majority of demands for most systems, making them more scalable, adaptable, and resilient [25].

A detailed descriptive model for each pattern is essential to performing a rigorous comparative analysis of IoT design patterns. This model serves as a structured profile that encapsulates the intrinsic characteristics of a pattern. Specifically, for every pattern discussed in this study, we propose the following key attributes:

– Typical Domain of Application determines whether the pattern is primarily suited for the Edge, Fog, Cloud, or hybrid architecture. This classification is crucial since IoT systems operate at different layers, each with unique constraints and performance requirements [26];

– Resource Requirements specify the demands on system resources such as memory and processing power. These parameters are particularly significant for IoT devices, which are often resource-constrained [27];

– Protocol Compatibility details the communication protocols (e.g., MQTT, CoAP, HTTP) with which the pattern is compatible. Given the heterogeneity of IoT networks, ensuring interoperability is a fundamental requirement [28];

– Impact on Latency assesses how the pattern affects communication delays, which are classified qualitatively as low, medium, or high. Latency is a crucial performance metric in time-sensitive IoT applications [29];

– Security Level evaluates how much the pattern incorporates security measures (e.g., encryption, authentication). This attribute reflects the pattern's ability to safeguard data integrity and confidentiality [16];

– Additional Constraints include any further limitations or prerequisites, such as the need for a specialized network environment or centralized management mechanisms.

The selection of specific attributes for the pattern description model is informed by a comprehensive analysis of IoT system requirements, as outlined in [14] and [15]. In contrast to traditional software pattern templates, which predominantly emphasize structural aspects, the proposed model prioritizes operational characteristics that directly influence the deployment of IoT systems. Existing models, such as the one presented in [16], often overlook essential considerations, including resource limitations and protocol compatibility – factors that are particularly critical in resource-constrained IoT environments.

By integrating both technical parameters, such as resource requirements and latency impact, and deployment-oriented aspects, including domain suitability and security level, the proposed model provides a holistic evaluation framework explicitly tailored to multi-tier IoT architectures. This enables a more granular and context-aware comparison of architectural patterns across diverse implementation scenarios.

The resulting pattern profiles serve as the foundation for quantitative evaluation, wherein each pattern's attributes are assessed using user-defined weight coefficients. These coefficients are applied within a weighted-sum model to compute an aggregated score for each pattern. This score provides an objective metric for ranking and comparing alternatives, thereby supporting rational and transparent design decision-making.

The final selection of design patterns is inherently a trade-off between technical requirements and project-specific resource constraints. For example, critical system nodes may require high-assurance security mechanisms, such as the use of the Secure Adapter pattern in conjunction with multi-level access control. In contrast, peripheral sensors might employ lightweight encryption schemes to conserve battery life.

Through these steps, the methodology delivers a structured and justifiable process for evaluating and selecting architectural patterns, incorporating both quantitative metrics and domain-specific priorities. It clarifies why a particular set of patterns is optimal for a given IoT system and ensures that selected solutions align with both functional demands and operational limitations. A representative set of evaluated patterns is provided in Table 1.

Table 1. **The analyzed set of software design patterns**

Pattern	Typical domain	Resource requirements	Protocol compatibility	Impact on latency	Security level	Implementation complexity
Secure Adapter [5]	Edge / Fog	Medium (requires encryption, but not critical)	MQTT/HTTP (with TLS) is supported	Medium	High (encryption, authentication)	Medium (requires additional configuration)
Secure Directory [5]	Fog / Cloud	Medium (central database or service)	Any (depends on the implementation)	Medium	High (stores keys, certificates, ACL)	Medium/High (requires deployment and administration)

Table 1 (continued)

Pattern	Typical domain	Resource requirements	Protocol compatibility	Impact on latency	Security level	Implementation complexity
Secure Logger [6]	Fog / Cloud	Medium (storage for logs)	Any (protocol-independent)	Low	Medium/High (tamper-evident logs)	Medium (requires centralized log storage)
Exception Manager [6]	Edge / Fog / Cloud	Low/Medium (depends on the complexity of error handling)	Any (protocol-independent)	Low	Medium (improves system resilience)	Medium (requires error categorization)
Reference Monitor [6]	Fog / Cloud	Medium (central access control)	Any (applied at API layer)	Medium	High (centralized policy enforcement)	Medium/High (requires policy management)
Access Matrix Authorization Rules [6]	Cloud	Medium (matrix storage)	Any (authorization layer)	Medium	High (fine-grained permissions)	Medium (requires permission matrix maintenance)
Input Validation Pattern [7]	Edge / Fog	Low (simple validation logic)	Any (applied at data entry points)	Low	High (prevents injection attacks)	Low/Medium (standard validation libraries)
Role-Based Access Control [8]	Fog / Cloud	Low (the main task is to manage role storage)	Any (depends on the authorization server implementation)	Low	High (flexible permission system)	Medium (requires role/account databases)
Token-Based Authorization Pattern [8]	Fog / Cloud	Medium (token generation and validation)	HTTP/REST with JWT is commonly used	Low / Medium	High (delegated authentication)	Medium (requires authorization server)
Client-Server [9]	Edge / Fog / Cloud	High (server), Low (clients)	HTTP, WebSockets, various protocols	Medium	Medium (depends on implementation)	Medium (standard architecture)
Peer-to-Peer [9]	Edge / Fog	Medium (decentralized operation)	Custom P2P protocols, WebRTC	Medium / High	Low/Medium (decentralized trust)	High (complex network topology)
Representational State Transfer [10]	Fog / Cloud	Low/Medium (stateless design)	HTTP	Medium	Medium (depends on implementation)	Low (widely adopted standards)
Publish-Subscribe [11]	Edge / Fog / Cloud	Low (lightweight MQTT brokers)	Best for MQTT; HTTP/webhooks are also possible	Low / Medium	Low/Medium (depends on additional security layers)	Low/Medium (ready-made libraries and brokers)
Monitor-Analyze-Plan-Execute-Knowledge [12]	Fog / Cloud	High (continuous monitoring and analysis)	Any (architecture-independent)	Medium	Medium (adaptive security possible)	High (complex control loops)
Sense-Compute-Control [13]	Edge / Fog	Medium (local processing)	Lightweight protocols preferred	Low	Low/Medium (depends on implementation)	Medium (requires coordinated components)

Table 1 (continued)

Pattern	Typical domain	Resource requirements	Protocol compatibility	Impact on latency	Security level	Implementation complexity
Observer / Controller Architecture [14]	Edge / Fog	Medium (requires organization of subscribers and controllers)	Can work with both MQTT and CoAP	Low / Medium	Low (typical architecture; security depends on the protocol)	Medium (requires setting up feedback mechanisms)
Singleton [15]	Edge	Low (minimizes resource usage)	Any (internal pattern)	Low	Low (not security-focused)	Low (simple implementation)
Cache-Aside [15]	Edge	Low/Medium (depends on cache size)	Any (mostly not tied to a specific protocol)	Low	Low (caching itself does not add security)	Low (many ready-made solutions)
Closed-Loop Control [16]	Edge / Fog	Medium/High (active data exchange, regulation)	MQTT/industrial protocols (Modbus, OPC UA)	Medium (potentially high for large volumes)	Low/Medium (again, it depends on specific protocols)	Medium (requires controllers, sensors, actuators)
Device-to-Device [17]	Edge	Low/Medium (P2P solutions between devices)	May require custom P2P protocols	Medium	Low (depends on encryption implementation)	Medium (mutual device authentication)
Cloud-in-the-Loop [18]	Cloud / Fog	Medium/High (depends on data volume and exchange frequency)	HTTP/REST, MQTT, gRPC, etc.	Medium / High	Medium (usually secure Cloud/Fog channels)	Medium/High (multi-level infrastructure)
Cloud-on-the-Loop [19]	Cloud / Fog	High (cloud-based decision making)	HTTP/REST, MQTT	High	Medium/High (cloud security)	High (complex infrastructure)
Device Discovery Pattern [20]	Edge / Fog	Medium (discovery mechanisms)	mDNS, UPnP, Bluetooth protocols	Medium	Medium (device authentication)	Medium (protocol implementation)
Data Processing Pattern [21]	Fog / Cloud	Medium/High (analytical processing)	Any (typically MQTT/HTTP)	Medium	Medium / High (sensitive data handling)	Medium/High (data analysis algorithms)
Service Composition Pattern [22]	Fog / Cloud	Medium/High (service orchestration)	HTTP/REST, SOAP, gRPC	Medium	Medium (service-level security)	Medium/High (service integration)
STL4IoT [23]	Edge / Fog	Medium (central coordination point)	Multiple protocols (protocol translation)	Low / Medium	Medium (central security point)	Medium (hub configuration)

Source: **compiled by the authors**

Employing a single formal template for describing IoT design patterns offers significant advantages. A unified structure enables consistent documentation, facilitating comparison, classification, and reuse of patterns across various application domains. This approach helps identify recurring themes and usage scenarios, thereby enhancing understanding and applicability. Moreover, a standardized format supports the development of automated tools for pattern identification, selection, and implementation, streamlining the design process and reducing the likelihood of errors. The adoption of a common representation also promotes more transparent communication and knowledge sharing among researchers, developers, and stakeholders. Consequently, the formalization of pattern descriptions not only strengthens the rigor of evaluation and selection processes but also accelerates real-world implementation. Ultimately, it contributes to greater innovation, scalability, and continuous improvement within IoT ecosystems.

4. METHODOLOGY FOR PATTERN SELECTION

When designing an IoT system, developers face a wide range of architectural patterns that address distinct concerns, including security, scalability, energy efficiency, edge-level data processing, fog-level interactions, and cloud-based analytics and control. Selecting the most appropriate combination of patterns for a specific system requires the application of a formal, structured methodology. The overall schema of the proposed methodology is illustrated in Fig. 1.

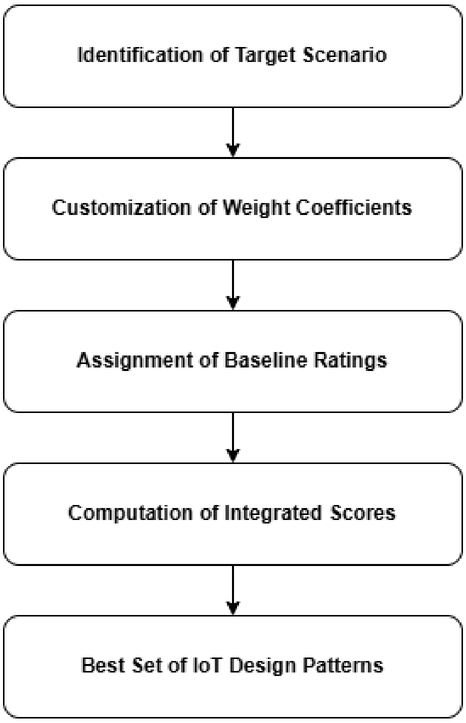

Fig. 1. **The schema of the pattern's selection methodology**
Source: **compiled by the authors**

To enable objective comparison among potential design solutions, it is essential to define relevant evaluation criteria and specify acceptable value ranges. A quality model tailored for IoT systems, as described in [30], serves as the foundation for this assessment. According to this model, key quality attributes include reliability, safety, usability, responsiveness, adaptability, durability, interoperability, and security.

To operationalize the quality model for pattern selection, qualitative characteristics are transformed into quantitative metrics using a five-point Likert scale. Each criterion is rated on a scale from 1 (minimal satisfaction) to 5 (maximum satisfaction). This quantitative framework provides a transparent and reproducible basis for comparing architectural patterns across various IoT system contexts. Each pattern is assigned numerical scores based on these criteria, derived from pattern descriptions and documented properties. For instance, the Secure Adapter pattern may receive a score of 5 for security and 3 for usability.

Based on these assessments (see Table 1 for pattern properties), we construct evaluation profiles for the analyzed set of patterns (Table 2). These profiles are derived from an extensive literature review and empirical analysis. It is important to note that the provided ratings are recommendations; developers are encouraged to adjust the scores based on domain-specific knowledge, experience, or project-specific requirements.

The evaluation profiles presented in Table 2 are based on a comprehensive literature review and empirical analysis of documented properties and typical use cases associated with each pattern. The resulting scores reflect consolidated expert assessments and are intended to serve as a baseline for comparison and evaluation. For example, the Role-Based Access Control pattern received a score of 5 in the Security category, attributed to its well-established effectiveness in enterprise systems [8]. Additionally, it scored 5 in Usability, supported by its widespread adoption and the availability of mature tooling.

To enable an objective comparison of alternative IoT design patterns, we utilize a quality evaluation table in which each pattern is assessed against a defined set of relevant quality criteria. Each criterion is assigned a weight coefficient that reflects its relative importance in the evaluation process. These weights, either expert-defined or recommended based on domain-specific best practices, are normalized such that the sum of all weights equals one.

Let r_i denote the quality rating (on a 1-5 Likert scale) for the i-th criterion, and let w_i be the corresponding normalized weight.

The aggregated score S for a given pattern is then computed as:

$$S = \sum_{i=1}^{n} (r_i \times w_i). \quad (1)$$

Table 2. **The evaluation profiles of software design patterns**

Pattern	Reliability	Safety	Usability	Responsiveness	Adaptability	Durability	Interoperability	Security
Secure Adapter	4	4	3	3	4	4	5	5
Secure Directory	5	4	3	4	2	5	2	5
Secure Logger	3	2	3	3	2	3	2	5
Exception Manager	5	4	4	4	3	5	2	4
Input Validation Pattern	5	5	4	5	4	5	3	5
Reference Monitor	5	5	2	3	3	5	2	5
Access Matrix Authorization Rules	4	4	2	3	2	3	2	5
Role-Based Access Control	5	4	5	5	5	5	4	5
Token-Based Authorization Pattern	5	3	5	5	5	5	5	5
Client-Server	4	3	5	4	3	4	4	4
Peer-to-Peer	4	3	2	5	5	4	3	2
Representational State Transfer	5	3	5	4	5	5	5	4
Publish-Subscribe	4	3	3	4	5	4	5	4
Monitor-Analyze-Plan-Execute-Knowledge	5	4	4	3	5	5	3	3
Sense-Compute-Control	3	3	3	3	4	3	4	3
Observer/Controller Architecture	5	5	3	4	5	5	2	2
Singleton	3	1	4	4	1	3	1	1
Cache-Aside	4	1	4	5	4	4	2	2
Closed-Loop Control	5	5	5	5	3	5	2	1
Device-to-Device	4	5	3	5	5	4	4	3
Cloud-in-the-Loop	2	2	4	2	5	3	5	3
Cloud-on-the-Loop	5	5	5	5	5	5	5	4
Device Discovery Pattern	4	3	5	4	5	5	5	3
Data Processing Pattern	4	2	4	4	5	5	5	3
Service Composition Pattern	3	2	5	3	5	4	5	3
STL4IoT	5	4	4	3	5	5	3	2

Source: **compiled by the authors**

Since the weights are normalized ($\sum_{i=1}^{n} w_i = 1$), the aggregated score represents a weighted average of the individual quality ratings. This score enables the ranking of design patterns and serves as a basis for decision-making when selecting the most suitable set of patterns for a specific IoT application.

Users may adopt the default weight coefficients tailored to a particular domain or customize them to reflect project-specific priorities and unique constraints. In both cases, the careful normalization of weights ensures that the resulting evaluations remain consistently reliable, comparable, and meaningful across different project contexts and scenarios.

A structured procedure comprising five sequential stages ensures transparent and reproducible selection of the best architectural patterns for IoT systems:

– Identification of Target Scenario. Selection of one predefined domain, e.g., industrial IoT, smart home, healthcare, or smart city. Each domain is associated with a set of weight coefficients that reflect characteristic priorities, including reliability, scalability, and energy efficiency;

– Customization of Weight Coefficients. Adaptation of weighting coefficients to the characteristics of the system being developed;

– Assignment of Baseline Ratings. Attribution of expert-derived ratings, if necessary, to each candidate pattern, stored in a pattern profile database (Table 1). Profile fields include typical latency impact, resource footprint, code complexity estimates, and security resilience indicators;

– Computation and Ranking. Calculate aggregated scores according to formula (1). Rank patterns in descending order of score to identify the most suitable options;

– Selection and Validation. Extraction of the top-N patterns and comparison of the recommended set against established best practices and documented case studies;

This methodology covers the entire process: from scenario selection and weight adjustment to rating application, final score calculation, and validation. The default configuration parameters serve as a starting point while maintaining full flexibility for adaptation to specific project requirements.

5. VALIDATION THROUGH CASE STUDIES

This section presents the application of the proposed methodology under various system configurations.

5.1. General-Purpose IoT System (Smart Home)

The first case study involves the development of a general-purpose smart home system, in which no specific quality attribute is prioritized. As a result, the weighting coefficients assigned to the evaluation criteria are approximately equal, as shown in Table 3. This balanced distribution reflects the absence of dominant concerns and serves as a representative baseline for evaluating architectural patterns in similar general-purpose smart home scenarios.

Table 3. **Normalized weights**

Quality criterion	Weight
Reliability	0.15
Safety	0.15
Usability	0.10
Responsiveness	0.15
Adaptability	0.10
Durability	0.10
Interoperability	0.15
Security	0.10

Source: **compiled by the authors**

All evaluated architectural patterns were ranked according to their computed aggregated scores, as presented in Table 4, and arranged in descending order. This ranking enables a straightforward comparison of the value of each pattern within the context of the defined weighting scheme.

Five top-ranked patterns with aggregated scores exceeding 4.0 were selected for further analysis: Role-Based Access Control, Cloud-on-the-Loop, Token-Based Authorization, Representational State Transfer, and the Device Discovery Pattern. This selection not only highlights the diversity of architectural approaches but also underscores the balanced consideration of security, interoperability, and performance requirements that guided the evaluation. The threshold value of 4.0 was chosen to distinguish between high and satisfactory quality levels, enabling a focus on the most appropriate solutions without introducing redundancy.

Table 4. **Aggregated scores and pattern ranking**

Pattern	Aggregated Score
Role-Based Access Control	4.75
Cloud-on-the-Loop	4.75
Token-based Authorization Pattern	4.70
Representational State Transfer	4.35
Device Discovery Pattern	4.35
Input Validation Pattern	4.30
Device-to-Device	4.20
Secure Adapter	4.00
Publish-Subscribe	4.00
Monitor-Analyze-Plan-Execute-Knowledge	3.95
Data Processing Pattern	3.95
Closed-Loop Control	3.95
Observer/Controller Architecture	3.90
STL4IoT	3.85
Exception Manager	3.85
Client-Server	3.85
Secure Directory	3.75
Service Composition Pattern	3.65
Reference Monitor	3.65
Peer-to-Peer	3.55
Sense-Compute-Control	3.25
Cloud-in-the-Loop	3.15
Cache-Aside	3.15
Access Matrix Authorization Rules	3.15
Secure Logger	2.90
Singleton	1.90

Source: **compiled by the authors**

To examine redundancy and compatibility among the selected patterns, a quality attribute matrix was constructed (Table 5). In this matrix, a pattern is marked with a “+” for each quality criterion where it achieved the maximum score (i.e., 5), based on the evaluation results from Table 2.

The final selection of architectural patterns involves not only identifying those with the highest scores but also ensuring that the selected patterns form a cohesive and non-redundant set. An examination of the top-ranked patterns, as presented in Table 5, highlights their complementary roles. Specifically, Role-Based Access Control and Token-Based Authorization collectively establish a robust foundation for system security. The Cloud-on-the-Loop pattern contributes to scalability and efficient system orchestration, whereas Representational State Transfer (REST) supports interoperability at the API level. Additionally, the Device Discovery Pattern addresses the dynamic behavior characteristic of smart home environments. Collectively, these patterns address critical architectural concerns, including security, scalability, interoperability, and responsiveness, thus providing a comprehensive architectural foundation.

Table 5. **Matrix of qualitative criteria for selected patterns**

Quality criterion	Reliability	Safety	Usability	Responsiveness	Adaptability	Durability	Interoperability	Security
Role-Based Access Control	+		+	+	+	+		+
Cloud-on-the-Loop	+	+	+	+	+	+	+	
Token-Based Authorization Pattern	+		+	+	+	+	+	+
Representational State Transfer	+		+		+	+	+	
Device Discovery Pattern			+		+	+	+	

Source: **compiled by the authors**

The results demonstrate a well-balanced set of architectural patterns designed to meet the diverse requirements of general-purpose systems. Each pattern serves a distinct function aligned with specific project priorities, such as security, compatibility, or responsiveness.

5.2. Industry 4.0 System

The second case study addresses the development of an Industry 4.0 system. The weighting coefficients assigned to the quality criteria are detailed in Table 6.

Table 6. **The quality criteria for Industry 4.0**

Quality criteria	Weights
Reliability	0.20
Safety	0.20
Usability	0.10
Responsiveness	0.15
Adaptability	0.05
Durability	0.15
Interoperability	0.10
Security	0.05

Source: **compiled by the authors**

In this context, Reliability and Safety are assigned the highest weights of 0.20 each, reflecting their critical importance in industrial environments, where system failures may result in substantial financial losses, safety hazards, and operational disruptions. These weightings are based on established best practices in industrial automation.

Responsiveness and Durability follow in priority, recognizing the need for timely reactions to dynamic operational conditions and ensuring long-term system stability. The remaining criteria are also considered, but with lower priority in conventional industrial settings, where functional robustness often takes precedence over flexibility and user-centric design.

Using the specified weights, aggregated scores were calculated and used to rank the architectural patterns. The top five patterns selected based on these scores are shown in Table 7.

Table 7. **Top-ranked patterns based on evaluation results**

Pattern	Aggregated Score
Cloud-on-the-Loop	4.95
Role-Based Access Control	4.70
Input Validation Pattern	4.65
Token-Based Authorization Pattern	4.60
Closed-Loop Control	4.40

Source: **compiled by the authors**

Notably, the Cloud-on-the-Loop and Closed-Loop Control patterns, both recognized in prior studies [31], also appear in the top-ranked results obtained through this methodology. This confirms the validity and practical alignment of the method with expert recommendations. Furthermore, the methodology identifies alternative patterns that satisfy the weighted quality requirements, demonstrating its utility in adapting to specific system demands.

5.3. Modified Industry 4.0 Configuration

To further demonstrate the flexibility of the methodology, we present a modified version of the Industry 4.0 system, where the primary objectives are shifted to emphasize Adaptability and Security. To accommodate this shift, the weights of these criteria are increased to 0.20 each, while the weights of other criteria are proportionally reduced (Table 8) to maintain the normalization constraint.

Table 8. **Modified quality criteria**

Quality criteria	Weights
Reliability	0.15
Safety	0.10
Usability	0.10
Responsiveness	0.10
Adaptability	0.20
Durability	0.10
Interoperability	0.05
Security	0.20

Source: **compiled by the authors**

This configuration prioritizes flexible and secure operations, possibly in contexts where systems are frequently reconfigured or exposed to external threats. However, the reduced emphasis on Interoperability (0.05), Responsiveness (0.10), and Reliability (0.15) implies trade-offs in seamless integration, rapid response, and fault tolerance.

Based on this updated weighting scheme, the top five patterns identified are: Role-Based Access Control, Token-Based Authorization, Cloud-on-the-Loop, Input Validation, and Representational State Transfer (Table 9). Their ranking is attributed to a strong alignment with the prioritized quality criteria: RBAC, Token-Based Authorization, and Cloud-on-the-Loop offer excellent adaptability and security; Input Validation contributes high reliability and durability; and Representational State Transfer supports scalable and interoperable component interaction.

Table 9. **Top-ranked patterns based on modified criteria**

Pattern	**Aggregated Score**
Role-Based Access Control	4.85
Token-Based Authorization Pattern	4.80
Cloud-on-the-Loop	4.80
Input Validation Pattern	4.60
Representational State Transfer	4.50

Source: **compiled by the authors**

These cases illustrate the methodology’s capability to adapt to varied system requirements by adjusting evaluation weights. The final set of patterns aligns well with both expert recommendations and modified system priorities, further confirming the effectiveness, flexibility, and extensibility of the proposed approach.

It should be noted that the proposed method addresses only the static contexts of architectural design. To ensure the consistency of the selected pattern sets in dynamic contexts, established verification techniques may be employed. For instance, the use of Petri nets to model and validate the dynamic behavior of an architecture composed of the top-ranked patterns can introduce an additional level of rigor into the design process [32].

CONCLUSIONS AND PROSPECTS OF FURTHER RESEARCH

The analysis of architectural patterns for IoT system development highlights their essential role in addressing key challenges such as security, scalability, and adaptability within distributed environments. The proposed method for selecting an optimal set of architectural patterns has been empirically validated, demonstrating both its effectiveness and practical applicability.

The developed approach offers significant value to IoT system architects and developers, particularly during the early stages of system design and development. It facilitates a systematic architectural decision-making process and provides objective justification for selecting specific patterns across diverse application domains. In the context of industrial IoT, for example, the method consistently emphasizes the benefits of employing patterns such as Closed-Loop Control, Cloud-in-the-Loop, and Publisher, which collectively enhance process reliability and support efficient data exchange.

Future research directions include expanding the architectural pattern knowledge base for emerging IoT domains, developing domain-specific pattern catalogs (e.g., for healthcare, smart cities, and agriculture), and integrating the proposed method into automated design environments.

REFERENCES

1. Atzori, L., Iera, A. & Morabito, G. “The Internet of Things: A Survey”. *Computer Networks*. 2010; 54 (15): 2787–2805. DOI: https://doi.org/10.1016/j.comnet.2010.05.010.

2. Ebrahim, N. S. “Complexity of IoT world – Review of challenges and opportunities in application development”. *Proceedings of the 2023 International Conference on Smart Computing and Application (ICSCA)*. 2023. DOI: https://doi.org/10.1109/ICSCA57840.2023.10087783.

3. Washizaki, H., Ogata, S., Hazeyama, A., Okubo, T., Fernandez, E. B. & Yoshioka, N. “Landscape of Architecture and Design Patterns for IoT Systems”. *IEEE Internet of Things Journal*. 2020; 7(10): 10091–10101. DOI: https://doi.org/10.1109/JIOT.2020.3003528.

4. Rajmohan, T., Nguyen, P. H. & Ferry, N. “Research Landscape of Patterns and Architectures for IoT Security: A Systematic Review”. *Proceedings of the 2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. 2020. p. 463–470, https://www.scopus.com/pages/publications/85096559490. DOI: https://doi.org/10.1109/SEAA51224.2020.00079.

5. Wen-Tin, L. & Po-Jen, L. "A Case Study in Applying Security Design Patterns for IoT Software System". *Proceedings of the IEEE International Conference on Applied System Innovation.* 2017. p. 1162–1165, https://www.scopus.com/pages/publications/85028536390.
DOI: https://doi.org/10.1109/ICASI.2017.7988402.

6. Ishfaq, A. & Muhammad, A. "Applying Security Patterns for Authorization of Users in IoT Based Applications". *Proceedings of the International Conference on Engineering and Emerging Technologies (ICEET)*. 2018, https://www.scopus.com/pages/publications/85050532496.
DOI: https://doi.org/10.1109/ICEET1.2018.8338648.

7. Fang, Z., Liu, Q., Zhang, Y., Wang, K. & Wang, Z. "IVDroid: Static Detection for Input Validation Vulnerability in Android Inter-component Communication". *Proceedings of the Information Security Practice and Experience*. 2015; 9065: 378–392. DOI: https://doi.org/10.1007/978-3-319-17533-1_26.

8. Ravidas, S., Lekidis, A., Paci, F. & Zannone, N. "Access Control in Internet-of-Things: A Survey". *Journal of Network and Computer Applications.* 2019; 144: 79–101.
DOI: https://doi.org/10.1016/j.jnca.2019.06.017.

9. Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M. & Ayyash, M., "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications". *Proceedings of the IEEE Communications Surveys & Tutorials*. 2015; 17 (4): 2347–2376. DOI: https://doi.org/10.1109/COMST.2015.2444095.

10. Rahman, L., Ozcelebi, T. & Lukkien, J. "Designing IoT Systems: Patterns and Managerial Conflicts". *Proceedings of the IEEE International Conference on Pervasive Computing and Communications Workshops*. 2019; 1: 542–548, https://www.scopus.com/pages/publications/85067944316. DOI: https://doi.org/10.1109/PERCOMW.2019.8730573.

11. Ajayi, O., Bagula, A., Bode, J. & Damon, M. "A Comparison of Publish-Subscribe and Client-Server Models for Streaming IoT Telemetry Data". *Emerging Technologies for Developing Countries*. 2023; 503, https://www.scopus.com/pages/publications/85171165256. DOI: https://doi.org/10.1007/978-3-031-35883-8_9.

12. Krupitzer, C., Temizer, T., Prant, T. & Raibulet, C. "An Overview of Design Patterns for Self-Adaptive Systems in the Context of the Internet of Things". *IEEE Access*, 2020; 8: 187384-187399, https://www.scopus.com/pages/publications/85102897357. DOI: https://doi.org/10.1109/ACCESS.2020.3031189.

13. Moin, A. "Sense-Deliberate-Act Cognitive Agents for Sense-Compute-Control Applications in the Internet of Things and Services". *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering.* 2015; 150: 23–28, https://www.scopus.com/pages/publications/84948769422. DOI: https://doi.org/10.1007/978-3-319-19656-5_4.

14. Kommuri, S., Rath, J. & Veluvolu, K. "Sliding-Mode-Based Observer–Controller Structure for Fault-Resilient Control in DC Servomotors". *IEEE Transactions on Industrial Electronics. 2018;* 65 (1): 918–929. DOI: https://doi.org/10.1109/TIE.2017.2721883.

15. Hurbungs, V., Bassoo, V. & Fowdur, T. "Software Design Pattern on the Edge". *Proceedings of the International Conference on Electrical, Computer, Communications and Mechatronics Engineering.* 2022; 1: 1–6. DOI: https://doi.org/10.1109/ICECCME55909.2022.9987912.

16. Shi, Y., Yi, C., Chen, B., Yang, C., Zhai, X. & Cai, J., "Closed-Loop Control of Edge-Cloud Collaboration Enabled IIoT: An Online Optimization Approach". *Proceedings of the International Conference on Communications*. 2022; 1: 5682–5687, https://www.scopus.com/pages/publications/85137273742. DOI: https://doi.org/10.1109/ICC45855.2022.9838906.

17. Pawar, P. & Trivedi, A. "Device-to-Device Communication Based IoT System: Benefits and Challenges". *IETE Technical Review*. 2018: 36 (4): 362–374. DOI: https://doi.org/10.1080/02564602.2018.1476191.

18. Maller, L., Suskovics, P. & Bokor, L. "Edge Computing in the Loop Simulation Framework for Automotive Use Cases Evaluation". *Wireless Networks*. 2023; 29: 3717–3735, https://www.scopus.com/pages/publications/85164019359. DOI: https://doi.org/10.1007/s11276-023-03432-3.

19. Kehl, P. et al. "Comparison of 5G Enabled Control Loops for Production". *Proceedings of the 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications.* 2020; 1: 1–6, https://www.scopus.com/pages/publications/85094132257.
DOI: https://doi.org/10.1109/PIMRC48278.2020.9217176.

20. Ben Hassine, M., Tounsi, I. & Kacem, M. "Developing IoT-based Smart Health Monitoring Systems Using Design Patterns". *Proceedings of 20th ACS/IEEE International Conference on Computer Systems and*

Applications. 2023; 1: 1–8, https://www.scopus.com/pages/publications/85190158728. DOI: https://doi.org/10.1109/AICCSA59173.2023.10479310.

21. Pramanik, P. K. D. & Choudhury, P. "IoT Data Processing: The Different Archetypes and Their Security and Privacy Assessment". *Internet of Things Security: Fundamentals, Techniques and Applications*. 2018; Chapter 3: 37–54, https://www.scopus.com/pages/publications/85065491336. DOI: https://doi.org/10.1201/9781003338642.

22. Safaei, A., Nassiri, R. & Rahmani, A. "Enterprise Service Composition Models in IoT Context: Solutions Comparison". *The Journal of Supercomputing.* 2022; 78: 2015–2042, https://www.scopus.com/pages/publications/85108604298. DOI: https://doi.org/10.1007/s11227-021-03873-7.

23. Rempillo, C. & Mustafiz, S. "STL4IoT: A Statechart Template Library for IoT System Design". *Simulation*. 2024; 101 (3): 213–239, https://www.scopus.com/pages/publications/105001571149. DOI: https://doi.org/10.1177/00375497241290369.

24. El-Afifi, M. I., Sedhom, B. E., Sanjeevikumar, P. & Eladl, A. A. "A Review of IoT-Enabled Smart Energy Hub Systems: Rising, Applications, Challenges, and Future Prospects". *Renewable Energy Focus*. 2024; 51: 100634. DOI: https://doi.org/10.1016/j.ref.2024.100634.

25. Borelli, F., Biondi, G., Horita, F. & Kamienski, C. "Architectural Software Patterns for the Development of IoT Smart Applications". *ArXiv*. 2020; abs/2003.04781: 1–29. DOI: https://doi.org/10.48550/arXiv.2003.04781.

26. Vo, T., Dave, P., Bajpai, G. & Kashef, R. "Edge, Fog, and Cloud Computing: An Overview on Challenges and Applications". *arXiv* Preprint arXiv:2211.01863. 2022. DOI: https://doi.org/10.48550/arXiv.2211.01863.

27. Perera, C., Zaslavsky, A., Christen, P. & Georgakopoulos, D. "Context Aware Computing for the Internet of Things: A Survey". *IEEE Communications Surveys & Tutorials*. 2014; 16 (1): 414–454. DOI: https://doi.org/10.1109/SURV.2013.042313.00197.

28. Yassein, M. B., Shatnawi, M. Q. & Al-Zoubi, D. "Application Layer Protocols for the Internet of Things: A Survey". *Proceedings of the 2016 International Conference on Engineering and MIS (ICEMIS)*. 2016. p. 7745303. DOI: https://doi.org/10.1109/ICEMIS.2016.7745303.

29. Pereira, C., Pinto, A., Ferreira Ferreira, D. & Aguiar, A. "Experimental Characterization of Mobile IoT Application Latency". *IEEE Internet of Things Journal*. 2017; 4 (4): 1082–1094, https://www.scopus.com/pages/publications/85029480278. DOI: https://doi.org/10.1109/JIOT.2017.2689682.

30. Liubchenko, V. & Chumachenko, D. "Quality Model for Software for Bionic Prostheses". *Information Technology: Computer Science, Software Engineering and Cyber Security*. 2023; 4: 51–57. DOI: https://doi.org/10.32782/IT/2023-4-6.

31. Bloom, G., Alsulami, B., Nwafor, E. & Bertolotti, I. "Design Patterns for the Industrial Internet of Things". *Proceedings of 14th IEEE International Workshop on Factory Communication Systems.* 2018; 1: 1–10. DOI: https://doi.org/10.1109/WFCS.2018.8402353.

32. Martynyuk, O. N., Drozd, O. V., Nesterenko, S. A. & Ahmesh, T. "Behavioral verification of internet of things systems by Petri nets". *Applied Aspects of Information Technology*. 2019; 2 (4): 295–303. DOI: https://doi.org/10.15276/aait.04.2019.4.

Conflicts of Interest: The authors declare that they have no conflict of interest regarding this study, including financial, personal, authorship or other, which could influence the research and its results presented in this article

Received 20.03.2025
Received after revision 28.05.2025
Accepted 10.06.2025

DOI: https://doi.org/10.15276/aait.08.2025.12
УДК 004.451.4

Системний підхід до вибору архітектурних патернів для розробки IoT

Чумаченко Данило Кирилович[1)]
ORCID: https://orcid.org/0009-0000-1477-534X; chumachdk@gmail.com

Любченко Віра Вікторівна[1)]
ORCID: https://orcid.org/0000-0002-4611-7832; lvv@op.edu.ua. Scopus Author ID: 56667638800
[1)] Національний університет «Одеська політехніка», пр. Шевченка, 1, Одеса, 65044, Україна

АНОТАЦІЯ

Зростання складності і масштабу систем Інтернету речей (IoT), особливо в промислових середовищах, створює значні виклики перед проєктуванням систем, включаючи питання безпеки, сумісності, масштабованості та ефективного використання ресурсів. Маючи у своєму розпорядженні широкий спектр архітектурних шаблонів для вирішення цих завдань, розробники часто постають перед труднощами при виборі найбільш придатних рішень. У цій статті представлена систематична методологія оцінки та вибору найкращих комбінацій архітектурних шаблонів проєктування, адаптованих до різних сценаріїв розгортання IoT. Підхід починається з аналізу наявних шаблонів проєктування IoT та моделювання їхніх ключових операційних характеристик. Для опису кожного шаблону використовується структурований шаблон, що сприяє узгодженості та порівнянності. Ці описи оцінюються за допомогою моделі якості, що включає такі критерії, як надійність, безпека, зручність використання, швидкість реагування, адаптивність, довговічність, сумісність та безпека. Модель зваженої суми з регульованими вагами критеріїв перетворює якісні оцінки на кількісні комплексні бали. Це дозволяє об'єктивно ранжувати шаблони та підтримує обґрунтоване прийняття архітектурних рішень. Методологія перевірена на основі численних прикладів, включаючи системи IoT загального призначення (наприклад, розумні будинки) та середовища Industry 4.0. У кожному випадку шаблони обираються на основі пріоритетів, специфічних для системи. Варто зазначити, що високопродуктивні шаблони, такі як Cloud-on-the-Loop, Closed-Loop Control та Role-Based Access Control, добре узгоджуються з відомими найкращими практиками та демонструють практичну застосовність методу. Аналіз чутливості додатково підтверджує адаптивність підходу, ілюструючи, як зміни в оцінках ваг значно впливають на кінцевий рейтинг шаблонів. Ця систематична методологія покращує відтворюваність, прозорість та гнучкість процесів проєктування архітектури IoT. Вона дає розробникам можливість адаптувати архітектурні рішення до конкретних потреб галузі, зберігаючи при цьому відповідність галузевим стандартам. Майбутні дослідження будуть спрямовані на розширення методології на нові сектори IoT, створення спеціалізованих каталогів шаблонів та інтеграцію системи вибору в автоматизовані інструменти проєктування для подальшої оптимізації розробки систем IoT.

Ключові слова: Інтернет речей; архітектурні шаблони проєктування; методологія вибору шаблонів; модель оцінки якості; архітектура системи

ABOUT THE AUTHORS

Danylo K. Chumachenko – PhD Student, Software Engineering Department. Odesa Polytechnic National University, 1 Shevchenka Ave. Odesa, 65044, Ukraine
ORCID: https://orcid.org/0009-0000-1477-534X; chumachdk@gmail.com
***Research field*:** IoT development; Android Engineering; Software System Srchitecture

Чумаченко Данило Кирилович – аспірант кафедри Інженерії програмного забезпечення, Національний університет «Одеська політехніка», пр. Шевченка, 1, Одеса, 65044, Україна.

Vira V. Liubchenko – Doctor of Engineering Sciences, Professor, Software Engineering Department. Odesa Polytechnic National University, 1 Shevchenka Ave. Odesa, 65044, Ukraine
ORCID: https://orcid.org/0000-0002-4611-7832; lvv@op.edu.ua. Scopus Author ID: 56667638800
***Research field*:** Software Engineering, Data Science, Project Management

Любченко Віра Вікторівна – доктор технічних наук, професор кафедри Інженерії програмного забезпечення, Національний університет «Одеська політехніка», пр. Шевченка, 1, Одеса, 65044, Україна