*Journal of Discrete Mathematical Sciences & Cryptography*
ISSN 0972-0529 (Print), ISSN 2169-0065 (Online)
DOI : 10.47974/JDMSC-2694

# An approach for IoT graph modeling for handling smart city services

Amit Bansal †
Siddhartha Sankar Biswas *
Md. Tabrez Nafis §
Suraiya Parveen ‡
Parul Agarwal @
Mohd Abdul Ahad #
*Department of Computer Science and Engineering*
*School of Engineering Sciences & Technology*
*Jamia Hamdard (Deemed to be University)*
*Hamdard Nagar*
*New Delhi 110062*
*Delhi*
*India*

**Abstract**

The Internet of Things (IoT) has laid the foundation for smart cities by allowing devices to work in synergy through focused automation. Nonetheless, most procedures available today are missing a holistic model that allows for clear visualization of how users are interacting with, how services are being effectively rendered, and how these procedures are scalable. To remedy this oversight, our generalized IoT graph model traces the flow of and advances an IoT action from inception, in direct response to a user's petition, to conclusion at a children's leaf node. Our model centrally manages user interaction through a Directory, or a digital page, for requests, which travel via 3 interconnected graph-layers (Top, Middle, and Bottom) whose distinct functions and adapts to secure, scalable, personalized networks in smart city applications. The model engenders novel field of research, which we have named IoT Algebra, leading to formal services analysis, composition, and fine-tuning by researchers. We review its real-world application and precedent.

***Subject Classification:*** *05C90, 68M10, 68R10, 90B10, 94C15.*

***Keywords:*** *IoT graph, Page book, Leaf nodes, IoT algebra, Smart city, Scalability, IIoT.*

† *E-mail:* **amit.bansal03@gmail.com**
* *E-mail:* **ssbiswas@jamiahamdard.ac.in (Corresponding Author)**
§ *E-mail:* **tabrez.nafis@jamiahamdard.ac.in**
‡ *E-mail:* **suraiya@jamiahamdard.ac.in**
@ *E-mail:* **pagarwal@jamiahamdard.ac.in**
\# *E-mail:* **aahad@jamiahamdard.ac.in**

## 1. Introduction

The internet of things (IoT) has revolutionized the uses of common devices by connecting them to the internet. It enables objects to understand their environment, distribute and receive data, and act upon them, resulting in improved control in hospitals, transportation systems and cities management [1, 2]. For smart cities, It brings innumerable benefits for City services efficiencies and citizens betterment. Modeling IIoT systems with graphs have been proved to be of use in comprehending their architecture and operation [3].

Despite this progress, IoT applications still seem too fragmented. Most approaches tend to address solely on devices, or data, or services in isolation, but they do not account for the processes during a user demand [4]. A unified view that emphasizes the user, while enabling scalability and transparent analysis, must be implemented.

Every IoT action begins with a person making a request. That request might work perfectly, partially work, or fail based on what happens with the device. To make IoT systems better and easier to study, we need a clear way to track this entire process from start to finish.

This paper introduces a generalized IoT graph model that shows the complete journey of an IoT action. We use graph theory to represent users, their requests, how those requests move through the system, and where they get executed across three connected layers.

The major contribution of this work covers a layered IoT graph model representing end-to-end execution of IoT actions, a structured Page Book-based user interaction mechanism with clear semantics, a formal graph-theoretic definition of the IoT model, introduction of IoT Algebra as a potential theoretical foundation for future research, and a scalability-oriented design suitable for large-scale smart city deployments.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 presents the proposed IoT graph architecture and its components. Section 4 provides a formal definition and scalability discussion. Section 5 concludes the paper and outlines future research directions.

## 2. Literature Review

Recent research has explored various aspects of IoT systems for smart cities and home automation. Several studies describe IoT architectures for managing domestic devices with capabilities for security and energy saving [5, 6]. Others put focus on a cost-effective, energy-conscious home automation assisted by cloud and mobile interface [7, 8]. There are some survey papers that reviewed the IoT computing on mobile phones, requirements for middleware and the overall IoT vision, including interoperability, privacy and cybersecurity issues [9, 10, 11]. Additional work has investigated IoT architectures in general and aspects of design process [12, 13]. Other papers provide an in-depth look on issues like applying graph modeling to IoT device lifecycle assessment and detecting vulnerabilities in traffic by neural networks [14, 15]. Graph-based approaches have shown promise for supporting dynamic expansion, fault isolation, and service coordination in urban IoT deployments [16]. Mathematical frameworks, particularly algebraic methods, are gaining attention for formal verification and optimization of IoT systems [17, 18]. Most existing frameworks are things-centric, data-centric, or service-centric, but they typically describe IoT at a high architectural level without explicitly modeling how a user's request flows from submission to execution on a device. Few models formally link individual users to their private execution points, which matters greatly in large smart cities with millions of users and devices. Our IoT graph model addresses these gaps by explicitly including user nodes, structured requests via Page Books, controlled communication paths, and private endpoints in one unified graph-based framework with isolation, security, and scalability built in, while also introducing IoT Algebra for systematic analysis and optimization.

## 3. Proposed Architecture of the IoT Graph

This section provides a description of the proposed IoT graph and its various layers along with the functioning of each of its components. This proposed gener-alized model can be adapted in any specific domain as per the requirement of the users.

### 3.1 *Layers of IoT Graph*

There are three Layers in our proposed model of IoT graph which are:

1. Front IoT Layer
2. Middle IoT Layer, and
3. Rear IoT Layer

The IoT graph is designed in such a way that the Front Layer cannot directly communicate with the Rear Layer and conversely, but via Middle Layer only. It is only the Middle Layer which can communicate with both Front Layer and Rear Layer. These three layers are introduced and explained below in the context of the proposed IoT graph modeling.

### 3.2 *Front IoT Layer*

This is where users interact with the system. It's set up like a star pattern with all user nodes connecting to a central point called CEO(Front) or CEO(F), which is basically a software controller that manages everything users do in this layer. Each user has their own node and sends service requests through something called a Page Book—think of it as a personal digital notebook on your phone or computer with ready-made forms for different services. Every user gets a unique five-character password (Source Code) to log in, and each service they want has its own five-character Destination Code that only they and the CEO(Rear) know about (refer to Figure 1). This setup keeps everything secure and prevents different users' requests from getting mixed up, even if they're asking for the same type of service.

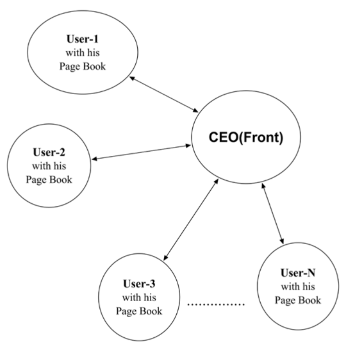


**Figure 1**

**A graph showing the Front IoT Layer**

### 3.3 *Page Book of a User*

The Page Book of a user consists of several pages. It is divided into two Chapters called by Chapter-1 and Chapter-2. The Chapter-1 consists of starting ten (can be adjusted) pages reserved for recording the valid list of services which can be sought by the concerned user. Chapter-2 consists of a large number of pages which are called by request pages. This page book can be opened by the user in his mobile phone device or in a computer, laptop etc. Different users have different Page Books of their own, and there is no scope for sharing of pages with other users.

3.4 *Chapter-1 and Registration of a User*

Every user interested to use IoT facility is required to register himself with an IoT System entering an Identity and a personal source code (password) of him, and then by entering a list of valid services of him (for example, we register ourselves with some internet agent like Yahoo, Gmail, Rediff mail, etc., to avail email facility) along with destination code for each service. The CEO(F) has the responsibility to complete his registration. The CEO(F) will authenticate the user identity and password (secret Source Code) of the user and also the secret Destination Code of each service item of the list of that user.

3.5 *Page Format in Chapter-1*

A page in Chapter-1 works like a simple table with four columns. The columns are Service Number, Service Description, Valid Parameter Values, and Destination Code. When a user registers, they create blank forms for their requests that include variable parts shown with quotes. Each variable can only take certain values that the user decides ahead of time. These values might be numbers, words, or simple choices. Users can add new services, remove old ones, or change the wording anytime they want.

**Table 1**

**A Page of a User in Chapter-1 after Registration**

| No. | Service Item | Valid Values | Code |
|---|---|---|---|
| 1 | In my bungalow my car garage door is to be ’X’ at time ’T’ Oclock. | X [Opened, Closed], T [0, 24]. | 8e2T% |
| 2 | My geyser must be ’X’ at time ’T’. | X [ON, OFF], T [0, 24]. | W936$ |
| 3 | Cooked rice must be ready in my kitchen room at ’T’ Oclock. | T [0, 24]. | 9841@ |
| 4 | My coffee must be ready at ’T’ Oclock” | T [0, 24]. | 8$5#+ |
| 5 | ........... | ............. | ....... |
| 6 | ........... | ............. | ....... |

### 3.6 *Chapter-2 of the User*

Consider a user, say User-1. In Page Book of User-1, there are several pages, but each page is only for one service to be sought by the User-1 from the IoT system. The list of valid services to be sought by User-1 is always known to User-1 because these are incorporated at the time of registration with option for modifications (insertion, deletion, editing, etc.).

### 3.7 *Page Format of Chapter-2*

Each of the above registered services sought by User-1 can be submitted using one page only of his Page Book. Two services cannot be requested using one single page. The format of pages is as below:

*Page (Chapter-2)*

1. Secret Source Code of the user = ——-
2. Secret Destination Code of the service = ——-
3. Request Serial Number = ———— (press enter)
4. Give inputs of each parameter: X = , T = , Y = , Z = .
5. Text Message, if any to be communicated at the leaf node:

On pressing 'Enter' after inputting at No. 3 in a page, the corresponding phrase from the Valid List (of Chapter-1) appears immediately here, and then inputs of parameters and text messages are to be typed. Text message will be communicated to the leaf node corresponding to the destination code of the request once this request is submitted to the IoT System.

### 3.8 *Illustrative Example*

Let's say User-1 has the source code %G5$% and wants coffee ready at home by 5 pm, but he's currently at his office in the morning. He opens his mobile app and fills out a page from Chapter-2. He enters his source code, selects service number 4 from his registered list (Table 1), which automatically shows "My coffee must be ready at 'T' O clock", and then he just types T = 17 for 5 pm. That's it—the request is submitted. Later, if he wants his garage opened at 4:30 pm, he opens another page, selects service 1, enters X = opened and T = 16.30, and submits. Even if two different users happen to request the exact same service, their unique source codes keep everything separate and prevent any mix-ups in the system.

3.9 *Middle IoT Layer*

The Middle Layer acts as a transmission bridge between Front and Rear parts. Information transferred between them, for example CEO(F) information flow to CEO(Rear), and other way round happen via this layer (Figure 2). This layer behaves like what happen on normal internet website browsing and emails.

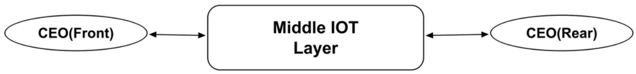


**Figure 2**

**Graph of the Middle Layer**

3.10 *Rear IoT Layer*

This is the area of actual IoT actions. The Rear Layer is built as a circle where CEO(Rear) is at the centre and connected to Manager nodes on the circle, and each Manager node connects to a batch of leaf nodes-the place where the services is running-on/off or a door to be opened-on. Each user has a private leaf node which are not shared; therefore, the system is safe, flexible and user could add or delete device freely (refer to Figure 3).

When something goes wrong at a leaf node, the Manager tells CEO(R), which reports back through the Middle Layer to CEO(Front), and finally the user gets notified. The secret codes keep everything secure, and all connections work both ways so leaf nodes can interact smartly with their Managers.

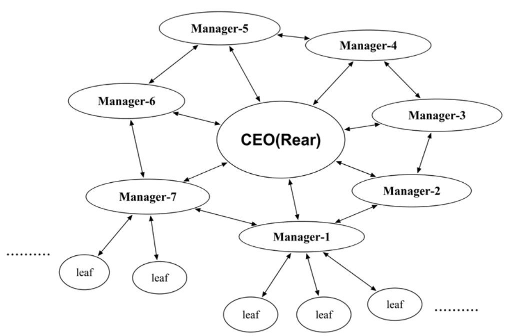


**Figure 3**

**Graph of Rear IoT Layer**

## 4. Proposed IoT Graph

Integrating the graphs of all the three layers we get the following graph (see Figure 4) for overall execution of an IoT action. This integrated graph is a general model of an IoT graph.

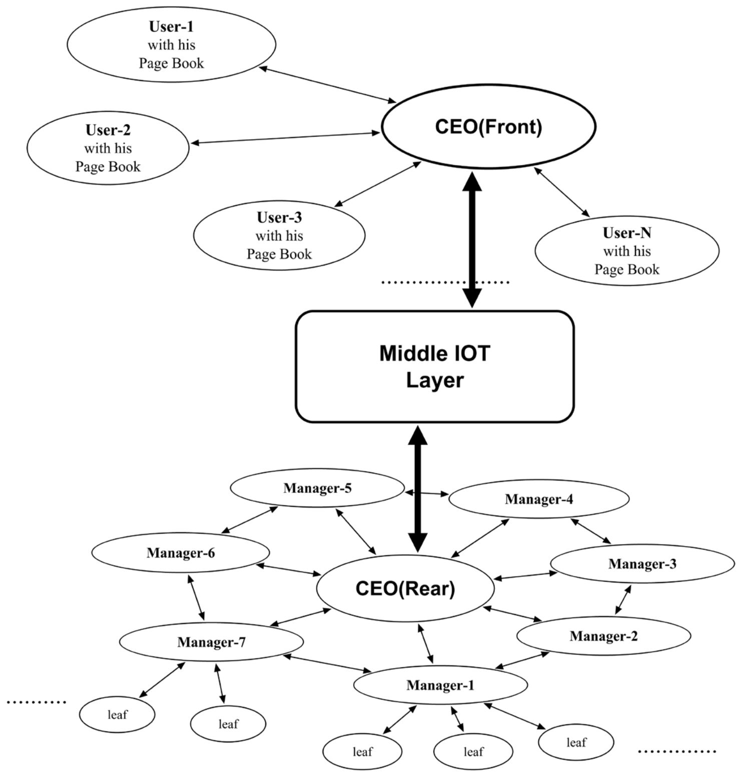


**Figure 4**

**IoT Graph: A general structure**

Every request of a user can be submitted from its own page book from any geographical location and the result of each request of him will happen at his leaves only. The Internet of things requires huge scalability in the network space to handle the surge of leaf nodes. Thus, in an IoT graph, the Front Layer uses star topology and the Rear Layer uses cycle topology [1].

### 4.1 *Formal Mathematical Definition of the Proposed IoT Graph*

Our IoT system is represented as a directed, labeled graph structure. We model it mathematically as a layered graph where the IoT graph $G_{IoT} = (V, E)$ contains vertices $V$ representing nodes and edges $E$ showing communication links between nodes. The vertices split into three separate groups for the three layers: $V = V_F \cup V_M \cup V_R$, where $V_F$ holds Front Layer nodes including users and CEO(Front), $V_M$ holds Middle Layer communication nodes, and $V_R$ holds Rear Layer nodes including CEO(Rear), Managers, and Leaf nodes. These three groups do not overlap. The Front Layer works as a star graph where user nodes $u_1, u_2, \dots, u_n$ and CEO(F) connect bidirectionally. Each user has a unique Source Code and services with Destination Codes. The Middle Layer handles message transfer between CEO(F) and CEO(R) without doing the actual work. The Rear Layer uses a cycle-based structure where CEO(R) connects to Manager nodes, and Managers connect to leaf nodes that execute the final actions. Direct communication between Front and Rear layers is not allowed, all messages must go through the Middle Layer following the path from user to CEO(F) to CEO(R) to Manager to leaf node. This keeps the system modular, secure, and scalable.

### 4.2 *IoT Algebra and Operations*

We now present IoT Algebra, a set of formal mechanisms to compute on IoT graph in an organized manner. It provides three fundamental operations: Service Composition(togethering services into larger workflows), Projection extracting out particular user for detailed study), and Reduction simplifying the graph by cutting down duplicate portions). All of these above operations satisfies several desired properties-if such operation ever produce valid IoT graph, enable services to execute independently, conceal user information-provide algebraic tools to label systems correctly, systematically archieving service administration and analyzing enormous IoT and IIoT systems.

### 4.3 *Applicability of the Proposed IoT Graph Model*

This IoT graph provides a clear representation of real smart city applications in a comprehensive way. It enables user/beneficiary interaction, communication paths, and execution endpoints as required in recently proposed IoT research [4]. Illustration - Smart Home Automation, Smart City Infrastructure, Simulation and Validation.

Overall, it provides an extensible approach of an extensible model to smart city IoT deployments.

## 5. Conclusion and Future Work

This paper proposed a generalized IoT graph model that can model the entire life cycle of user-involved IoT actions in a smart city in three interconnected layers as the Front, Middle, and Rear layers. Explicit modeling of users, services and endpoints as graph constructs overcomes a significant shortcoming of the current IoT architecture, which suffers from lack of formal execution modeling [4]. The formal mathematical formalism that we introduced has significant analytical applications and presents the IoT Algebra, a novel theoretical framework for reasoning about service composition, optimization, and scalability. Future work includes formal verification of IoT Algebra properties, methods to automate service composition, and scaling up the model for IIoT, where reliability and real-time control are factors. Our proposed architecture is a solid theoretical and architectural foundation for designing smart city IoT systems.

## References

[1] P. Sun, Y. Wan, Z. Wu, and Z. Fang, "A survey on security issues in IoT operating systems," *J. Netw. Comput. Appl.*, vol. 231, Art. no. 103976 (2024), doi: 10.1016/j.jnca.2024.103976.

[2] H. N. Dai, Z. Zheng, and Y. Zhang, "Blockchain for Internet of Things: A survey," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8076–8094 (2019), doi: 10.1109/JIOT.2019.2920987.

[3] S. Dargaoui, M. Azrour, A. El Allaoui, A. Guezzaz, A. Alabdulatif, and A. Al-Najim, "Internet of Things authentication protocols: Comparative study," *Comput. Mater. Contin.*, vol. 79, no. 1, pp. 1279–1304 (2024), doi: 10.32604/cmc.2024.048912.

[4] J. J. P. Abadía, C. Walther, A. Osman, and K. Smarsly, "A systematic survey of Internet of Things frameworks for smart city applications," *Sustain. Cities Soc.*, vol. 83, Art. no. 103949 (2022), doi: 10.1016/j.scs.2022.103949.

[5] L. Salman, S. Salman, S. Jahangirian, M. Abraham, F. German, C. Blair, and P. Krenz, "Energy efficient IoT-based smart home," in *Proc.*

*IEEE 3rd World Forum Internet Things (WF-IoT)*, Reston, VA, USA, pp. 526–529 (2016), doi: 10.1109/WF-IoT.2016.7845460.

[6] G. M. Sung, Y. S. Shen, J. H. Hsieh, and Y. K. Chiu, "Internet of Things-based smart home system using a virtualized cloud server and mobile phone app," *Int. J. Distrib. Sens. Netw.*, vol. 15, no. 9, Art. no. 1550147719879354 (2019), doi: 10.1177/1550147719879354.

[7] S. Jadon, A. Choudhary, H. Saini, U. Dua, N. Sharma, and I. Kaushik, "Comfy smart home using IoT," in *Proc. Int. Conf. Innov. Comput. Commun. (ICICC),* Adv. Intell. Syst. Comput., vol. 1165, pp. 495–504 (2020), doi: 10.1007/978-981-15-5148-2_43.

[8] S. K. Vishwakarma, P. Upadhyaya, B. Kumari, and A. K. Mishra, "Smart energy efficient home automation system using IoT," in *Proc. 4th IEEE Int. Conf. Internet Things: Smart Innov. Usages (IoT-SIU)*, Ghaziabad, India, pp. 1–4 (2019), doi: 10.1109/IoT-SIU.2019.8777479.

[9] A. Kamilaris and A. Pitsillides, "Mobile phone computing and the Internet of Things: A survey," *IEEE Internet Things J.,* vol. 3, no. 6, pp. 885–898 (2016), doi: 10.1109/JIOT.2016.2600569.

[10] M. A. Razzaque, M. Milojevic-Jevric, A. Palade, and S. Clarke, "Middleware for Internet of Things: A survey," *IEEE Internet Things J.,* vol. 3, no. 1, pp. 70–95 (2016), doi: 10.1109/JIOT.2015.2498900.

[11] S. Madakam, R. Ramaswamy, and S. Tripathi, "Internet of Things (IoT): A literature review," *J. Comput. Commun.,* vol. 3, no. 5, pp. 164–173 (2015), doi: 10.4236/jcc.2015.35021.

[12] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Comput. Netw.*, vol. 54, no. 15, pp. 2787–2805 (2010), doi: 10.1016/j.comnet.2010.05.010.

[13] P. P. Ray, "A survey on Internet of Things architectures," *J. King Saud Univ. Comput. Inf. Sci.,* vol. 30, no. 3, pp. 291–319 (2018), doi: 10.1016/j.jksuci.2016.10.003.

[14] M. Ramadane, S. Meyer, and D. Bohnet, "Environmental impact assessment of IoT devices: A graph-based approach," *Procedia Comput. Sci.,* vol. 236, pp. 338–347 (2024), doi: 10.1016/j.procs.2024.05.031.

[15] M. Gao, L. Wu, Q. Li, and W. Chen, "Anomaly traffic detection in IoT security using graph neural networks," *J. Inf. Secur. Appl.,* vol. 76, Art. no. 103532 (2023), doi: 10.1016/j.jisa.2023.103532.

[16] D. Goyal, A. Kumar, P. Mathur, F. Sheth, and A. K. Gupta, "Robust cybersecurity through discrete and large language models for effec-

tive phishing attack detection," *J. Discrete Math. Sci. Cryptogr.,* vol. 28, no. 5A, pp. 1621–1638 (2025), doi: 10.47974/JDMSC-2162.

[17] A. Noonia, D. Thakral, P. Mathur, F. Sheth, H. Shaikh, and A. K. Gupta, "A discrete mathematical model and cryptography for secure medical image analysis: Encrypted chest X-ray classification," *J. Discrete Math. Sci. Cryptogr.,* vol. 28, no. 5A, pp. 1473–1486 (2025), doi: 10.47974/JDMSC-2146.

[18] R. Joshi, P. Mathur, A. K. Gupta, S. Singh, V. Paliwal, and S. Nayar, "Mathematical modeling of intelligent system for predicting effectiveness of premenstrual syndrome," *J. Interdiscip. Math.,* vol. 26, no. 3, pp. 551–562 (2023), doi: 10.47974/JIM-1699.

*Received December, 2025*