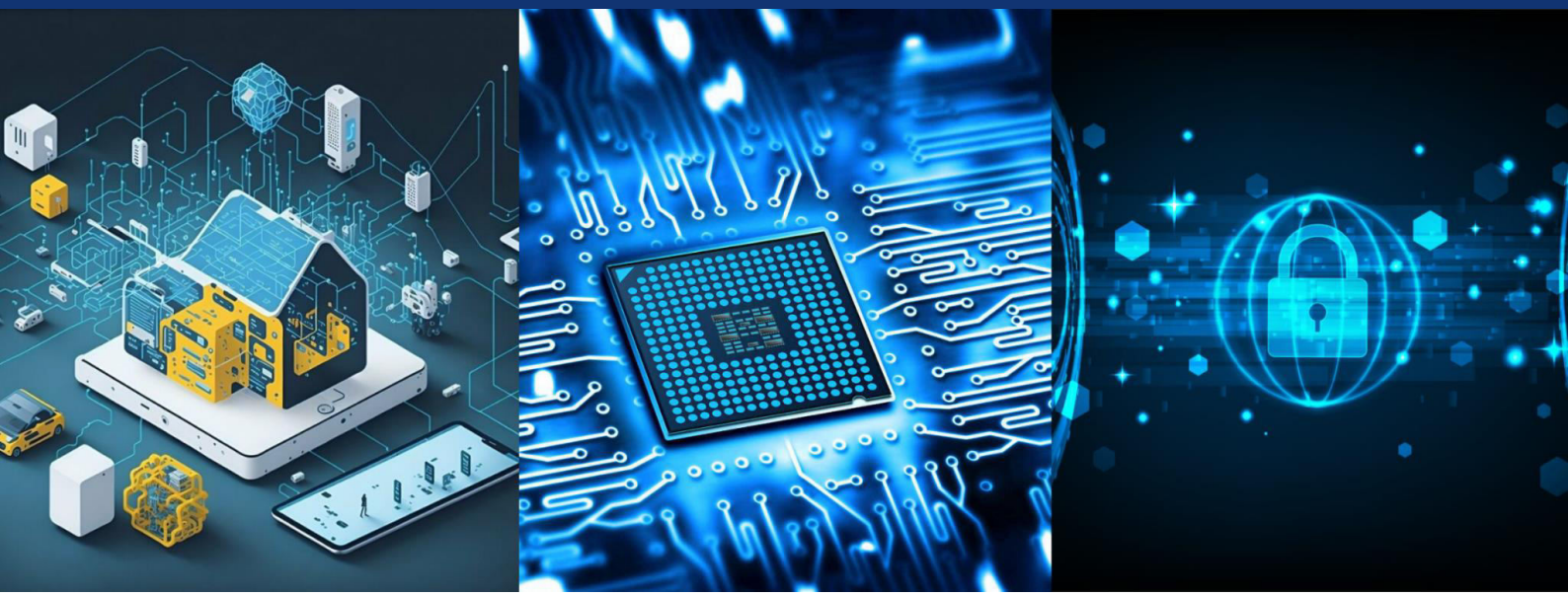
IJIRCCE

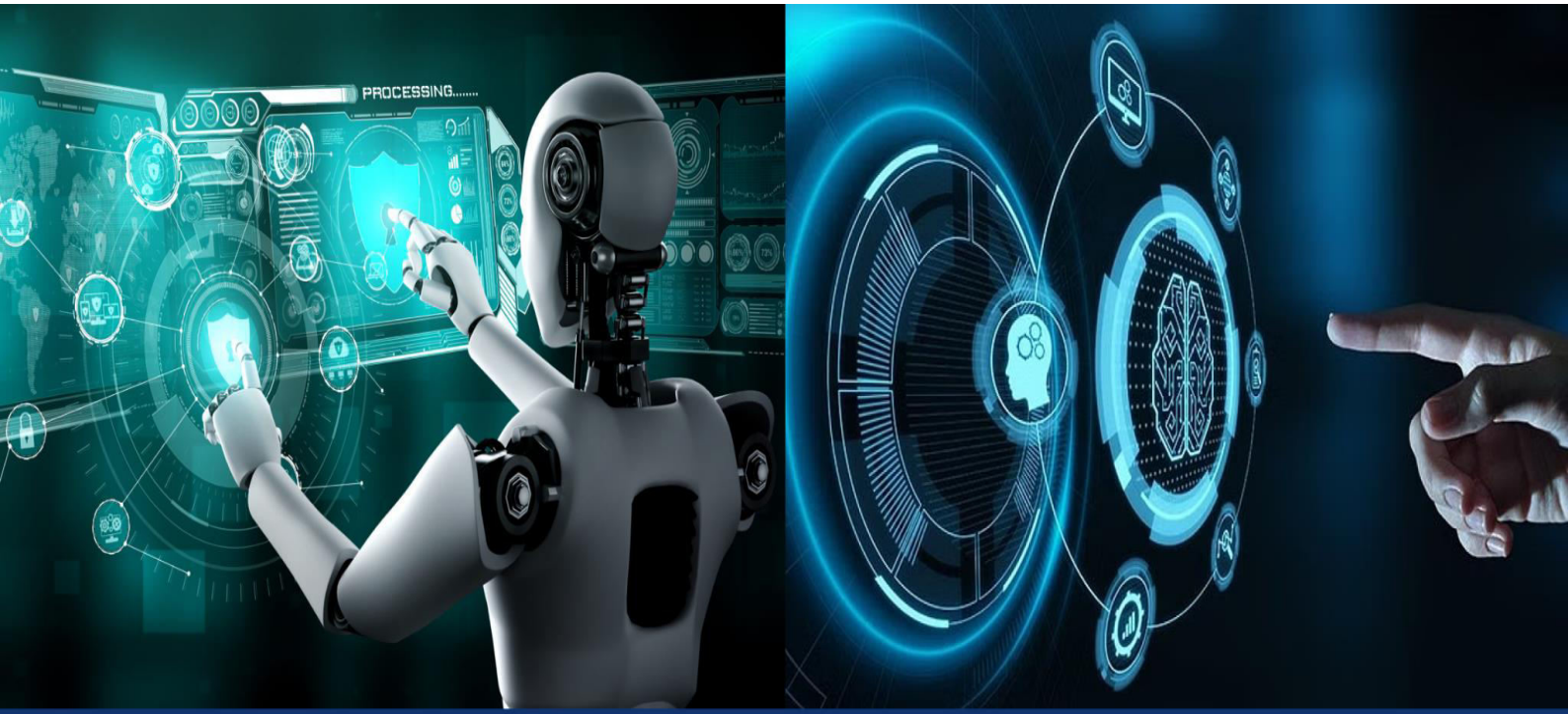
ISSN(O): 2320-9801

ISSN(P): 2320-9798



International Journal of Innovative Research in Computer and Communication Engineering

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 9.927

Volume 14, Issue 10, October 2026

© 2026 IJIRCCE | Volume 14, Issue 10, October 2026 DOI: 10.15680/IJIRCCE.2026.1410041

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Scalable IOT Device Intelligence with Adaptive Edge-Enabled Multi-Tenant AI Platform

Basavaraj[1], Manoj D[1], Pruthviraju G[1], Bhimashankar[1], Prof. Rubithra A[2]

UG Students, Dept. of Computer Science and Design, Rajarajeshwari College of Engineering, Bengaluru, Karnataka, India[1]

Assistant Professor, Dept. of Computer Science and Design, Rajarajeshwari College of Engineering, Bengaluru, Karnataka, India[2]

ABSTRACT: For organizations that depend on hardware with sensors, they require constant monitoring, but if they have to create a monitoring stack for each organization, then it is costly and slow to add new organizations. In this paper, we discuss an environment where the edge node is the tenant's laptop on which the platform is multi-tenant. It sends data from the laptop to a shared Supabase (PostgreSQL) backend, which receives data from an Arduino sensor rig (light intensity, temperature, voltage and device status), and performs light preprocessing The fact that every tenant's records are kept separate with a central AI worker that has access it needs to analyze all devices is a huge plus for Row Level Security (RLS). This worker uses a rolling mean, std, z score and fixed range checks to filter incoming readings, assesses the severity of each anomaly, and requests a language model to explain the event in a natural language. Readings, insights and alerts are sent to tenant and administrator web dashboards in real time via subscriptions. A small number of tenant laptops were used end to end with the prototype and large scale and quantitative evaluation are future work. The author and publisher are not liable for the use of these terms. The author and the publisher accept no liability for any liability arising from the use of these terms

I. INTRODUCTION

Now, a number of connected sensors are placed in buildings, laboratories, workshops and small factories and are measuring and reporting temperature, light level, supply voltage, etc., at all times. If no one, or nothing, is watching these streams are of no value. Raw numbers on a console cannot indicate to an operator if the drifting voltage is a normal noise or not, as well as indicating if it is a precursor to failure. For the small organization it can be difficult to justify the effort needed to establish the collection, storage, analysis and visualization. One such solution is to install a monitoring stack for each customer. This isolation of data comes with the drawback of having to manage another set of servers, databases and upgrades for each additional customer. The other way round centralization of all the raw readings spares duplication but does create significant amounts of unfiltered data being pushed over the network and leaves the data open to the possibility of one customer's data being seen by another. There's a middle ground between these extremes, which is a shared platform that is isolating tenants at the data layer and preparing some data near the hardware. This paper introduces such a platform. A typical laptop with a small Python edge agent is configured to each tenant to communicate with an Arduino based sensor rig. The agent reads the serial readings and writes it to a common Supabase project. PostgreSQL Row Level Security guarantees that a signed in tenant can only access their own data (devices, readings, insights and alerts). A separate tests, and , if it seems to be misread, then queries a language model that generates an intelligible explanation. The results are displayed in real time on Web dashboards created with Next.js, both for tenants and for an administrator.

This work has four contributions:

- A working multi-tenant design that combines the acquisition of data from the edge devices using laptops with a shared cloud database, separation of tenants within the cloud database.
- A lightweight analysis pipeline where a statistical screening determines the need for a language model call, rather than having explanations coupled to it.
- Live update tenant/administrator dashboards, along with subscriptions to reflect the number of devices connected.
- Real hardware demonstration of the entire chain from the hardware sensor to dashboard alert

© 2026 IJIRCCE | Volume 14, Issue 10, October 2026 DOI: 10.15680/IJIRCCE.2026.1410041

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

II. RELATED WORK

Platforms with multiple isolation rooms. Ding et al. [1] examine the tradeoff between isolation and efficiency when running multiple workloads in the same cloud, as they've seen machine learning become increasingly adopted to handle the new workloads like serverless functions and AI pipelines. Khayyam [2] addresses policies for controlling access and identity across distributed components, especially when policies are enforced across a variety of software as a service components that are powered by an AI. Thota [5] suggests an AI-powered SaaS approach to dynamically allocate workloads among Platform tenants using predictive analytics and reinforcement learning, while Sudhakar [7] elaborates on the architectural decisions to be made for multi-tenant AI platforms such as allocating resources to each tenant and tuning them for various performances requirements.

Intelligent Edge! Both Asmita [3] and Payal [8] explore the synergy between AI and eedge computing in the context of IOT, nothing advantages in terms of reduced latency, bandwidth consumption and privacy concerns when processing datain the edge. In Model optimization and efficient inference on resource constrained devices; in Edge AI architectures and deployment strategies, Gill et al. [9] provide a taxonomy and review of Edge AI architectures and deployment strategies. Shi et al. [11] give an overview of the vision and open challenges of edge computing . Manage the use of resources and secure computing, Kumar [4] suggests a predictive and policy driven mechanism to provide the balancing of resources across multiple clusters before imbalance is set up and Hemamalini and Mishra [10] discuss the integration of AI, Integrate blockchain and IOT into cloud apps for automating cloud security and resource management.

Anomaly detection. Deviation from expected behavior based statistical anomaly detection is a well established method which has been studied by Chandola et al. [12]. PostgreSQL [13] Provides a practical approach to impose the tenant boundaries close to the data through row level access control. Positioning the research listed above primarily deals with a single issue, either isolation, edge intelligence, or clud resource management. The focus here is on bringing these hardware, a managed database with built in tenant policies, and an analysis stage that relies on statistics, but only calls a language model when necessary. The integration is its contribution as well as the working prototype.

III. THE PROPOSED SYSTEM AND ALGORITHM INCLUDE THE FOLLOWING

Design Considerations:
Each tenant has an individual identity and each device, reading, insight and alert has that identity. The data is acquired by the sensor in the tenant laptop and the parsing and formatting process is done before the data leaves the site. Isolation is not limited to application code, but is enforced by the database policies. The AI worker is a trusted service that has cross tenant read, but all output that is written by the AI worker is attached to the AI worker's originating tenant and device. The language model is only called out in flagged events to reduce the number of times it has to be called, and to ensure that the explanations are based on the evidence that has been detected. Changes to dashboards are pushed and made real time no manual refresh required

© 2026 IJIRCCE | Volume 14, Issue 10, October 2026 DOI: 10.15680/IJIRCCE.2026.1410041

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

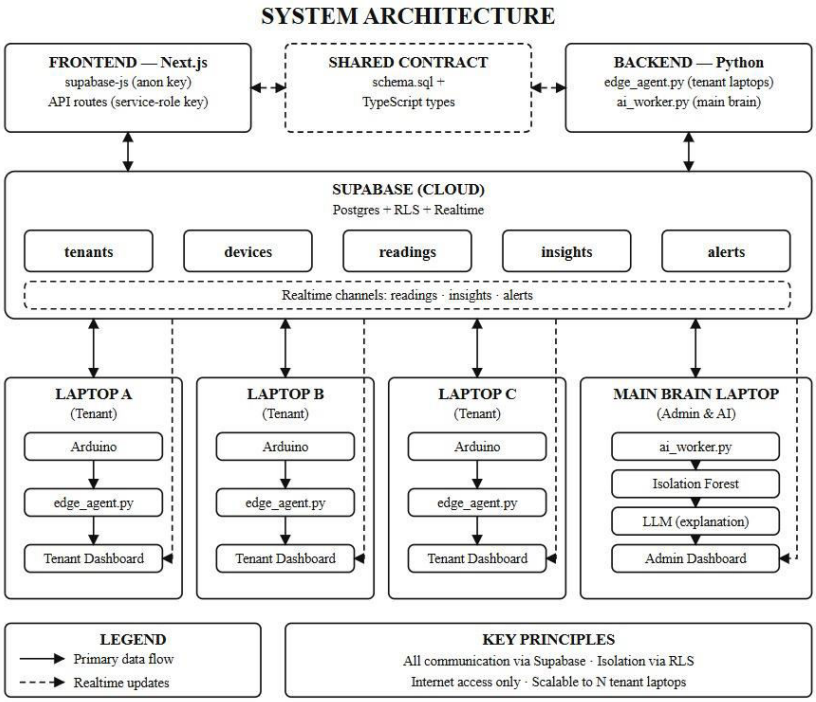


Figure 1 Architecture

This section gives the description of the proposed system. This is the description of the proposed system. The workflow involves acquiring data from the edge, ingesting the clouds, isolating the data and then presenting it, all of which are performed using AI analysis.

Step 1: Edge acquisition the Arduin rig collects temperature, light intensity, voltage and a flag indicating the state of the device and sends the information to the tenant's laptop. The edge agent reads every line, checks it for validity and parses it, it then converts it to a structured record and adds the device and tenant identifiers.

Step 2: Ingest and isolate clouds. This record is written to the readings table in Supabase by the agent. There are five primary tables in the schema: tenants, devices, readings, insights and alerts. Comparing the signed in user to the tenant column in each row, RLS policies allow a tenant to see only their own data when they run a query, but give an administrator role access to all data across all tenants.

Step 3: AI analysis. The AI worker follows new readings, implements a rolling window for each device/parameter, and performs the tests that are specified in Section III-B. if a reading is flagged, a severity grade is assigned to it and the language model is prompted to provide a description of what is going on in plain text based on the flagged value, the recent window statistics and the device context.

Step 4: Presentation. The rows of insights and alerts are stored back to Supabase Readings, insights and alerts are pushed to the tenant dashboard and administrator dashboard, all in real time in real time channels, which present live charts, device status and active and resolved alerts.

The same tenant identity is used when dealing with subscription handling a tenant registers and selects the subscription based on the number of devices they wish to connect, then add devices to the subscription up to the limit.

Anomaly Detection: Detects anomalies and grades them as a severity level. Let the set of the last n readings be the rolling window and let xt denote the latest reading for a fixed device and parameter x. The mean and standard deviation of the data in a window.

© 2026 IJIRCCE | Volume 14, Issue 10, October 2026 DOI: 10.15680/IJIRCCE.2026.1410041

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

$\mu t = (1/n) \Sigma i\ xi$, $\sigma t = \sqrt{[\ (1/n) \Sigma i\ (xi - \mu t)2\ ]}$,
The average of the new reading and the standard deviation of the new reading are given
$zt = (xt - \mu t) / \sigma t$
Reading is flagged if $|zt|$ is above a certain threshold value that can be set by user τ, or if xt is outside a user defined operating range [Lmin,Lmax] of that parameter. The fixed range check will detect abrupt failures that a rolling window will slowly absorb while the z-score check will detect drifting that will not exceed the nominal range. The flagged readings will have a grade of severity, which increases as the size of the deviation gets larger, and will be increased if the set limit is exceeded.

Processing pipeline of the AI worker. Language model is only called when reading is flagged; normal readings are stored only

IV. PSEUDO CODE

1. The algorithm 1 is running on each tenant laptop (Edge Agent) Load Tenant ID, device ID and cloud credential
2. Ensure serial port to the arduino is opened Repeat until agent is running
3. Read a line of sensor output
4. Parse the line ; ignore it if it is not well formed
5. Create a log file including temperature, light, voltage, status and date and time.
6. For step 6, attach the tenant ID and device ID
7. Insert record in readings table
8. If the upload is unsuccessful, make note and repeat later.

Stop the serial connection when 8. Close the serial connection is reached

1. The central machine has AI worker (algorithm 2) running. Use the service credential to connect to the database.
2. Make sure to get inserts for readings on the reading table
3. As you read each time you encounter a new word
4. Retrieve the n most recent readings of the same device and parameter
5. Compute e μ, σ and z using Eq. (1).
6. If $|z|>\tau$ or r is not within [Lmin, Lmax] then flag r as anomalous.
7. If r is not anomalous go to the next reading
8. Give a severity grade from the devialtion
9. Reason with the language model about an explanation add a tenant ID and device ID row of insights and alert row
10. When readings return to normal, mark alert as resolved

V. RESULTS AND DISCUSSION

It has been constructed and operated as a full end to end solution, Arduino sensors, tenant's laptops with the edge agent, tenant's shared Supabase project, the central AI worker and the web dashboards. The following three tenant laptops and one central laptop were used as illustrated in Fig 1. The evaluation was functional in nature, that is, to verify that each stage functions as it is intended to and that the stages interact A summary of the functional verification can be found in table 1 capability observed behavior Edge acquisition and preprocessing the sensor lines were parsed, formatted and uploaded to the edge by the edge agent. Records stored in the cloud with the correct tenant and device, tagged as tenants' Each tenant's session only returned the tenant's rows (Tenant isolation (RLS)) Anomaly Detection out of pattern and out of range readings were detected . AI generated insights Flagged events made comprehensible explanations with severity Live dashboards New readings, insights and alerts not refreshed Admin view Multiple tenants' devices consolidated in a single console

© 2026 IJIRCCE | Volume 14, Issue 10, October 2026 DOI: 10.15680/IJIRCCE.2026.1410041

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

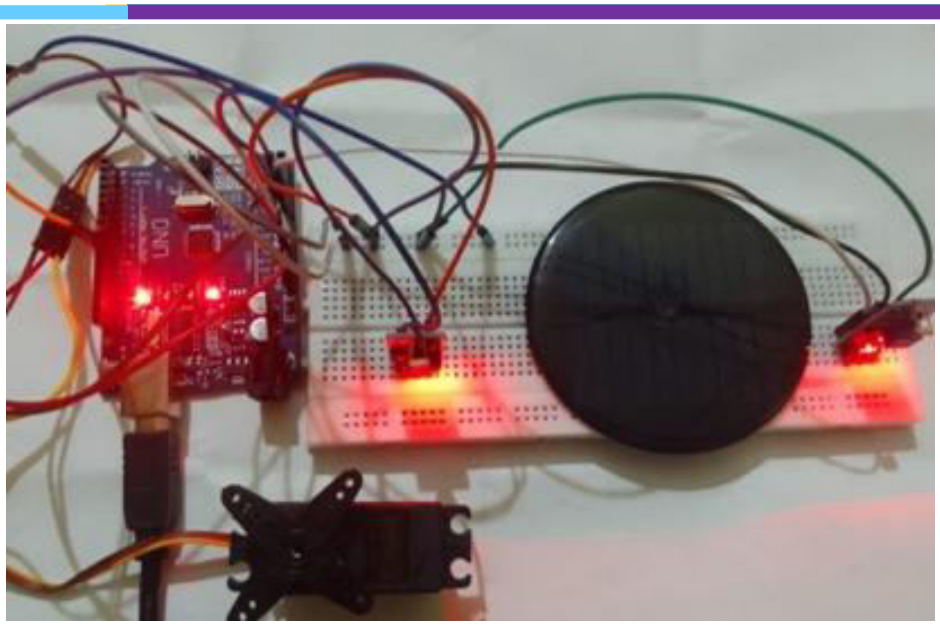


Figure 2 Physical Hardware Setup



Figure 3 Homepage of the Software

Prototype screens and hardware: Tenant dashboard, Admin dashboard, Insights and alerts, Hardware set up All modules demonstrated as per design in the demonstration environment as depicted in table 1. The statistical stage works with only a small number of recent readings and the language model is invoked only for flagged events, which means that the explanation step only runs as often as flagged events occur. Signing in as various tenants, tenant separation was verified to confirm that both tenants onlyu saw their respective devices, readings, insights and alerts

Limitations. The prototype has been tested with a small number of devices and tenants, and has not been tested with a large number of devices and tenants running simultaneously. No quantitative latency and throughput measurements are reported as this was not the main part of this phase. Analysis is currently the centerpiece, and therefore requires an active internet connection to synchronise, analyse and to allow for live updates, before and anomaly reaches the worker

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 9.927| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

and insight can be produced. The analysis stage is not industry specific and only has some user management for advanced users. Currently the platform can only monitor and alert does not actualte does not actuate devices

VI. CONCLUSION AND FUTURE WORK

This paper presented a multi tenant IOT monitoring platform composed of Arduino based sensors, laptop edge nodes, a common Supabase back end with Row Level Security, a statistics first AI worker and live dashboard. A working prototype was built that showed the entire journer of a physical read giving a severity based natural language alert to the relevant tenant on their dashboard, and obscuring all other tenant information. The Laptop is the edge node, so small organizations do not need to utilize dedicated gateway hardware to create the edge node, reducing the barrier to entry.

There are a number of extensions planned. Measurement of latency and throughput and detection quality to more tenants and devices comes first. Part of the analysis would be on the edge node, thus allowing for anomalies to be identified during network downtime. Fixed thresholds can be replaced or complemented by classifiers and predictive models that are derived from machine learning. Other types of sensors can be registered in the current device model. With appropriate security closed loop control of connected equipment can be added to the mix while the subscription layer can be enhanced with billing and more account management features.

REFERENCES

[1] J. Ding, T.Chen and Y.Qin, “Resource isolation in multi tenant cloud platforms without performance degradation”, Journal of computing and electronic information management 2026
[2] Khayyam, “Resilient identity and access governance architecture for artificial intelligence enabled software as a service ecosystem, “Saudi Journal of Engineering and Technology , vol. 11, no. 4,pp.2026,2026
[3] N.Asmita, “AI in edge computing and IOT”, International Journal of Innovative Reasearch in Technology, vol.12, no.11, pp.2026,2026
[4] B.Kumar, “AI based cloud resource optimization for multi cluster environments, “International Journal of Computer Applications,2025
[5] R.C.Thota, “Next generation multi tenant SaaS with AI orchestrated workload isolation for scalable performance”, world Journal of Advanced Engineering Technology and Sciences, vol. 16, no.2 ,2025
[6] S.O.Semerikov Edge intelligence unleashed: Deploying AI models in resource constrained environment Journal of Edge Computing =, 2025
[7] N.R.Sudhakar, “Multi tenant AI platforms Architecture for performance and efficiency”, The Cognitive Nexus, vol.1, no.2,25
[8] S. N. Payal, “AI in edge computing and IoT systems,” International Journal of Scientific Research in Engineering and Technology, vol. 11, no. 2, 2025.
[9] S. S. Gill et al., “Edge AI: A taxonomy, systematic review and future directions,” Cluster Computing, 2024.
[10] V. Hemamalini and A. K. Mishra, “Artificial intelligence-blockchain-enabled-internet of things-based cloud applications for next-generation society,” Automated Secure Computing, 2024.
[11] W. Shi, J. Cao, Q. Zhang, Y. Li and L. Xu, “Edge computing: Vision and challenges,” IEEE Internet of Things Journal, vol. 3, no. 5, pp. 637–646, 2016.
[12] V. Chandola, A. Banerjee and V. Kumar, “Anomaly detection: A survey,” ACM Computing Surveys, vol. 41, no. 3, Art. no. 15, 2009.
[13] PostgreSQL Global Development Group, “PostgreSQL documentation: Row Security Policies,”.



INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi crossref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details