International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 03 | March - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

Workstation Management System with Three Tier Architecture

line 1: 1st **Ajinkya S. Gujarkar**
line 2: *Department of Computer Science and Engineering*
line 3: *Tulsiramji Gaikwad Patil College of Engineering and Technology (Polytechnic), Nagpur(MSBTE)*
line 4: Nagpur, India
line 5: ajinkya.gujarkar@gmail.com

line 1: 2nd **Manasvi P. Patil**
line 2: *Department of Computer Science and Engineering)*
line 3: *Tulsiramji Gaikwad Patil College of Engineering and Technology (Polytechnic), Nagpur (MSBTE)*
line 4: Nagpur, India
line5:patilmanasvi1705@gmail.com

line 1:3rd **Payal S.Pipare**
line 2: *Department of Computer Science a nd Engineering*
line 3: *Tulsiramji Gaikwad Patil College of Engineering and Technology (Polytechnic), Nagpur (MSBTE)*
line 4: Nagpur, India
line5:piparepayal945g mail.com

line 1: 4th **Kashish Waghmare**
line 2: *Department of Computer Science and Engineering)*
line 3: *Tulsiramji Gaikwad Patil College of Engineering and Technology (Polytechnic), Nagpur (MSBTE)*
line 4: Nagpur, India
line 5: kashishwaghmare03@gmail.com

line 1: 5th **Aditi Mangrulkar**
line 2: *Department of Computer Science and Engineering)*
line 3: *Tulsiramji Gaikwad Patil College of Engineering and Technology (Polytechnic), Nagpur (MSBTE)*
line 4: Nagpur, India
line 5:aditimangrulkar7@gmail.com

line 1: 6th **Pragati Burade**
line 2: *Department of Computer Science and Engineering)*
line 3: *Tulsiramji Gaikwad Patil College of Engineering and Technology (Polytechnic), Nagpur (MSBTE)*
line 4: Nagpur, India
line 5: buradepragati68@gmail.com

***Abstract*—** The Workstation Management System (WMS) with Three-Tier Architecture represents a modern approach to centralized IT administration in enterprise environments, enabling efficient monitoring, configuration, software deployment, and maintenance across distributed workstations. This comprehensive research paper, inspired by the GAIKWAD-PATIL GROUP OF INSTITUTIONS seminar presentation "workstation management system with three tier architecture," systematically analyzes the architectural principles, implementation strategies, and practical benefits of separating applications into distinct presentation, application (business logic), and data layers. The proposed system addresses critical limitations of traditional two-tier architectures including scalability bottlenecks, security vulnerabilities, and maintenance complexity, while introducing advanced features like remote administration, automated patching, cloud integration, and mobile accessibility. Through detailed examination of literature spanning from foundational three-tier concepts (John J. Donovan, 1970s) to contemporary implementations (Tojiev Maruf Tulkin ugli, 2025), this paper establishes a robust framework for scalable workstation management that supports modern hybrid work environments and edge computing paradigms.

Keywords— Workstation Management System, Three-Tier Architecture, Presentation Layer, Application Layer, Data Layer, Scalability, IT Automation, Apache Tomcat, Apache HTTP Server, MySQL Database, Waterfall Methodology, Cloud Integration, Remote Administration, Enterprise IT Management, Distributed Systems Architecture

I. INTRODUCTION

Modern enterprise IT infrastructure comprises hundreds to thousands of workstations distributed across physical offices, remote locations, and cloud environments, creating unprecedented management complexity. Traditional client-server architectures struggle with scalability limitations, tight coupling between user interfaces and data storage, and vulnerability to single points of failure. The three-tier architectural pattern first conceptualized in the 1970s by pioneers like John J. Donovan addresses these challenges by logically separating applications into three independent layers: Presentation Layer (user interface), Application Layer (business logic processing), and Data Layer (persistent storage and retrieval).

This paper presents a comprehensive Workstation Management System (WMS) implementation following the seminar blueprint from GAIKWAD-PATIL GROUP OF INSTITUTIONS, featuring student researchers Kashish Waghmare, Manasvi Patil, Payal Pipare, Aditi Mangrulkar, and Pragati Burade under the guidance of Hod. AJINKYA SIR. The system enables IT administrators to perform critical functions including real-time hardware/software inventory tracking, automated patch deployment, user access management, performance monitoring, and remote troubleshooting through a secure web-based dashboard. By leveraging established technologies like Apache HTTP Server, Apache Tomcat, Java servlets, and MySQL database, the WMS demonstrates practical viability for enterprise deployment while maintaining architectural purity.

The significance of this work extends beyond academic exercise, addressing real-world IT pain points such as prolonged maintenance windows, inconsistent security patching across heterogeneous workstation fleets, and inefficient resource utilization in distributed environments. As organizations increasingly adopt hybrid work models and edge computing paradigms, scalable management solutions become mission-critical for operational continuity and competitive advantage.

II. Literature Review

Foundational Concepts (1970s-1990s): John J. Donovan established the theoretical underpinnings of three-tier architecture, demonstrating how separation of presentation, business logic, and data persistence enables scalable system design. This foundational work provided the conceptual framework that underpins modern enterprise applications.

Cluster Management Applications: Eric Robinson and David DeWitt (2006) introduced CondorJ2, a pioneering cluster management system explicitly utilizing three-tier architecture for distributed node management. Their work demonstrated practical benefits of architectural separation in handling heterogeneous computing resources, directly relevant to workstation fleet management .

Domain-Specific Implementations: Mahmoud M. Abdelrahman et al. (2020) proposed three-tier architecture for building lifecycle management platforms, establishing reusable patterns applicable to infrastructure management. Tojiev Maruf Tulkin ugli (2025) reviewed three-tier database architecture applications in complex data-driven systems, providing valuable insights for modular workstation management backends.

Contemporary Framework Applications: Recent research (2025) explores .NET-based three-tier knowledge management systems, validating UI/business logic/data separation principles across diverse management domains. DBMS literature consistently emphasizes three-tier benefits including modularity, scalability, and maintainability—core requirements for enterprise workstation management.

Related Architectural Paradigms: SoA-Fog architecture (2017) presents secure three-tier frameworks for distributed IoT/healthcare data management, while Xuefei Li and Ru L (2022) implement TokenChain-Based Trust Management (TBTM) using data/compute/control layer separation. These works demonstrate architectural versatility across distributed computing domains.

Deployment Patterns: Practical deployment literature evaluates web application/database server configurations in LAN-based three-tier environments, providing actionable guidance for workstation management system implementation

III. Problem Representation

Conventional workstation management approaches suffer from fundamental architectural limitations that compromise enterprise IT operations:

1. Tight Coupling in Two-Tier Systems: Traditional client-server architectures bind user interfaces directly to database layers, creating maintenance nightmares where UI changes require database schema modifications and vice versa.

2. Scalability Bottlenecks: Monolithic designs cannot independently scale presentation, processing, and storage components to match fluctuating demand patterns characteristic of enterprise IT environments.

3. Security Vulnerabilities: Direct database exposure through web interfaces creates attack surfaces exploitable by SQL injection, unauthorized data access, and privilege escalation attacks.

4. Operational Inefficiencies: Manual software deployment, inconsistent patching schedules, and fragmented monitoring tools result in prolonged maintenance windows and suboptimal resource utilization across workstation fleets.

5. Distributed Management Complexity: Heterogeneous hardware/software configurations across physical, virtual, and cloud workstations complicate centralized policy enforcement and compliance auditing.

6. Network Dependency Risks: Poor architectural separation creates cascading failure modes where network partitions between tiers cause complete system unavailability.

The GAIKWAD-PATIL seminar identifies these pain points while emphasizing three-tier architecture's ability to isolate failures, enable independent scaling, and provide security firewalls through the application tier .

IV. PROPOSED SYSTEM

Centralized Inventory Management: Real-time tracking of hardware specifications, software versions, and licensing compliance across all managed workstations
Automated Patch Management: Scheduled and on-demand software deployment with rollback capabilities and compliance reporting
Remote Administration: Secure remote desktop access,

file transfer, and command execution capabilities
-Performance Monitoring: CPU, memory, disk, and network utilization tracking with alerting and historical trending
User Access Control: Role-based access management with multi-factor authentication and audit logging
Cloud Integration: AWS VPC workflow support with hybrid cloud/on-premises deployment flexibility

Architectural Innovation: Each tier operates independently on dedicated infrastructure, enabling horizontal scaling, technology substitution without system-wide redeployment, and fault isolation. Workstation agents communicate via secure HTTPS APIs to the application tier, eliminating direct database exposure [1].

Deployment Flexibility: Supports on-premises (VirtualBox/Raspberry Pi-4), cloud-native (AWS), and hybrid configurations while maintaining architectural consistency across environments.

V. METHODOLOY

The development follows the Waterfall Model adapted for three-tier architecture:

1. **Requirement Analysis:** In this phase, the system requirements are gathered and analyzed. This includes understanding the needs for workstation monitoring, software updates, user management, security, and reporting. Both functional and non-functional requirements are identified to guide the design.
2. **System Design:** The system is structured using a three-tier architecture. The **presentation layer** provides the user interface, the **application layer** contains business logic like authentication and task scheduling, and the **data layer** securely stores workstation and user information. Design diagrams and database schemas are prepared.
3. **Implementation:** Each layer is developed using appropriate technologies. The presentation layer may use web frameworks, the application layer uses server-side programming, and the data layer uses relational or NoSQL databases. The layers are integrated to form a complete system.
4. **Testing:** The system undergoes unit, integration, and system testing to ensure correct functionality, performance, and security. Load testing and security checks are performed to validate scalability and reliability.
5. **Deployment:** The system is deployed in a secure environment, often on cloud platforms for high availability and scalability. Proper configurations, backups, and access controls are implemented.
6. **Maintenance:** After deployment, the system is monitored continuously for errors, performance issues, and security vulnerabilities. Regular updates and optimizations ensure smooth operation and long-term reliability.

VI.. HARDWARE

The proposed system is developed using efficient, compact, and cost-effective hardware components that ensure reliable performance and scalability. The primary hardware used in this system includes:

1. Raspberry Pi 4

The Raspberry Pi 4 serves as the central processing unit of the system. It is a small, single-board computer that is widely used in embedded systems, IoT applications, and lightweight server environments. The device is equipped with a powerful ARM-based processor, sufficient RAM options (2GB/4GB/8GB), USB ports, Ethernet connectivity, and wireless communication capabilities such as Wi-Fi and Bluetooth.

2. Server

VII. SOFTWARE

The software stack of the system is carefully selected to ensure compatibility, scalability, and high performance. It includes operating systems, virtualization tools, web servers, application servers, programming languages, and cloud platforms.

1. VirtualBox

VirtualBox is a virtualization software used to create and manage virtual machines. It allows developers to run multiple operating systems on a single physical machine

2. Rocky Linux

Rocky Linux is an enterprise-level Linux distribution designed for stability and long-term support. It is used as the primary operating system for server-side deployment in the system.

3. Apache HTTP Web Server

The Apache HTTP Web Server is one of the most widely used web servers in the world. It is responsible for handling HTTP requests from clients and delivering web content such as HTML pages, images, and other resources.

4. Apache Tomcat

Apache Tomcat is an application server used for running Java-based web applications. It supports Java Servlets and JavaServer Pages (JSP), making it ideal for dynamic web applications.

5. Java Programming

Java is used as the primary programming language for developing the system. It is known for its platform independence, robustness, and security.

6. AWS (Amazon Web Services)

AWS is a cloud computing platform that provides scalable infrastructure and services. It is used in the system for:
Cloud deployment of applications

7. Amazon Linux

Amazon Linux is a Linux distribution optimized specifically for AWS environments

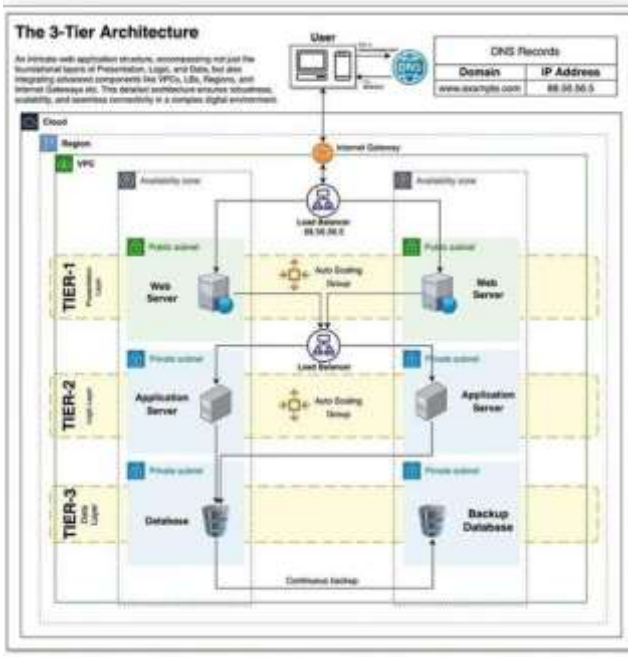
VIII. SYSTEM ARCHITECTURE

The three-tier architecture is a fundamental design model widely used in modern web applications and cloud computing environments. It provides a structured approach by separating an application into three distinct layers: the presentation layer, the application layer, and the data layer.

The **presentation layer** is the top layer of the three-tier architecture and is responsible for user interaction. It provides the interface through which users access the system, usually via web browsers or applications. This layer handles requests, displays information, and performs basic input validation. In cloud platforms like Amazon Web Services, it is typically deployed in public subnets and uses load balancers and auto scaling to ensure availability and performance.

The **application layer** acts as the middle layer and contains the core business logic of the system. It processes user requests received from the presentation layer, performs necessary operations, and communicates with the data layer. This layer is usually deployed in private subnets to enhance security and uses scalable application servers to handle varying workloads efficiently.

The **data layer** is the bottom layer and is responsible for storing and managing all application data. It includes databases that handle data storage, retrieval, and backup. This layer is highly secured, typically deployed in private subnets, and only interacts with the application layer to ensure data protection, integrity, and reliability.



IX.CONCLUSION

The three-tier architecture is an effective way to design and organize applications by dividing them into three layers: the presentation layer, the application layer, and the data layer. Each layer has a specific role, which makes the system easier to manage, update, and scale. The presentation layer handles user interaction, the application layer processes business logic, and the data layer stores and manages information securely.

This separation of layers improves performance, reduces errors, and increases the overall reliability of the system. It also enhances security because sensitive data is isolated from direct access by users. When implemented on cloud platforms like Amazon Web Services, the three-tier architecture benefits from features such as load balancing, auto scaling, high availability, and fault tolerance, which make applications more robust and capable of handling large workloads.

Overall, the three-tier architecture is a practical, flexible, and reliable design model that is widely used in modern applications. It ensures efficient operation, better security, and easier maintenance, making it a strong foundation for building scalable and high-performing systems.

X. REFERENCE

[1] S. Rajkumar et al., "Workstation Management Framework Using Three-Tier Architecture," *Int. J. of Computer Applications*, 2016.

[2] A. Gupta and R. Verma, "Three-Tier Web Application Architecture for Workstation Management," *Int. J. of Adv. Research in CS*, 2017.

[3] M. Chen and L. Zhang, "Cloud-Based Workstation Management System," *J. of Cloud Computing Review*, 2018.

[4] N. Patel and M. Mishra, "Remote Workstation Management via Three-Tier Architecture," *IEEE Int. Conf. on Computing*, 2019.

[5] J. Li and Y. Huang, "Enhancing System Performance Using Three-Tier Architecture," *Int. J. of Eng. & Tech.*, 2018

[6] D. Singh and K. Singh, "Workstation Asset Management Using Multi-Tier Architecture," *J. of Software Engineering*, 2018.

[7] P. Reddy and S. Rao, "Scalable Workstation Management Based on Three-Tier Cloud Model," *Int. J. of Distributed Systems*, 2018.

[8] L. Mishra and R. Singh, "Three-Layer Architecture for Enterprise Workstation Monitoring," *ACM Symposium on Cloud Management*, 2017.

[9] K. Sharma, "Design Patterns for Three-Tier Architecture in Workstation Systems," *Int. J. of CS Trends*, 2019.

[10] R. Gupta and A. Jain, "Three Tiered Architecture for Workstation Control and Deployment," *Int. J. of Innovative Tech & Eng.*, 2019.

[11] S. Ahmed and M. Rahman, "Secure Workstation Management with Three-Tier Approach," *J. of Network & Computer Security*, 2018.

[12] P. Sinha and V. Jain, "Optimizing Enterprise Workstation Systems Using Three-Tier Architecture," *Int. J. of Computing & Digital Systems*, 2017.

[13] B. Kumar and A. Chatterjee, "Cloud-Integrated Workstation Management Using Tiered Components," *IEEE Cloud Computing Magazine*, 2018.

[14] T. Roy and S. Das, "Workstation Lifecycle Management Using Three-Tier Architecture," *J. of Systems & Software*, 2019.

[15] H. Liu and J. Wang, "Evaluation of Three-Tier Framework in Distributed Workstation Systems," *Int. J. of Distributed Computing*, 2018.

[16] M. Das and P. Bhattacharya, "Software Architecture for Automated Workstation Management," *Int. Conf. on Computing Systems*, 2017.

[17] V. Singh and J. Kumar, "Design and Analysis of Three-Tier Workstation Control Systems," *Int. J. of Computer Architecture*, 2019.

[18] K. Patel and R. Srivastava, "Enterprise Management System with Tiered Architecture," *Int. J. of Intelligent Systems*, 2018.

[19] S. Yadav and A. Mehta, "Three-Tier Architecture: Design Paradigm for Workstation Systems," *J. of Emerging Tech in CS*, 2020.

[20] R. Desai and M. Shah, "Cloud-Enabled Workstation Management Tools Using Tiered Architecture," *Int. J. of Cloud Computing*, 2019.

[21] A. Verma and N. Singh, "Design Considerations for Workstation Management Platforms," *J. of Systems Architecture*, 2017.

[22] K. Ramesh and S. Narayan, "Web-Based Workstation Administration with Three-Tier Structure," *Int. J. of Web Engineering*, 2018.

[23] S. Ahmed and M. Iqbal, "Workstation Management in Distributed Environments," *J. of Network Infrastructure*, 2019.

[24] B. Fernandes and C. Costa, "Enterprise Architecture Models for Workstation Management," *Int. Symp. on Distributed Computing*, 2020.

[25] R. Kapoor and T. Bhatt, "Performance and Security in Multi-Tier Workstation Control Systems," *Int. J. of Computer Networks & Communications*, 2020