

# Design, Implementation, and Management of An Enterprise Voice Over IP (VOIP) Infrastructure

<sup>1</sup>P. Rakesh and <sup>2</sup>Dr. D. Shobha Rani

<sup>1</sup>PG Scholar, Dept; of CSE, Chadalawada Ramanamma Engineering College (Autonomous), Tirupati.

<sup>2</sup>Professor, Dept; of CSE, Chadalawada Ramanamma Engineering College (Autonomous), Tirupati.

**Abstract:** *Session Initiation Protocol (SIP) infrastructures are commonly used in Voice over IP (VoIP), unified communications, and real-time communication systems. Traditional SIP setups often depends on closely linked authentication systems and large database designs, which can limit scalability, flexibility, and ease of deployment. This paper introduces a scalable SIP communication structure built with the open-source Flexisip proxy server. It integrates with an external SQL database for managing users and routing. The new design reduces the number of internal authentication modules, allowing centralized user management, distributed routing logic, and effective handling of SIP messages. The system is designed with scalability, easy administration, effective handling of multiple SIP registrations and call sessions as priorities. Experimental analysis shows that this architecture supports large-scale SIP deployments while keeping latency low and using resources efficiently.*

**Keywords:** Flexisip, SIP, VoIP, SQL Database, Scalable Architecture.

## I. INTRODUCTION

Voice over IP (VoIP) technologies have become really important for businesses and real-time apps these days. Sip handles most of the signaling for calls and registrations. A lot of setups still depend on built-in authentication that does not scale well when things grow. This project looks at a different approach using the open source Flexi SIP server together with an external SQL database. It seems the idea is to move user data and routing decisions outside the proxy itself so management gets simpler. I think that helps with larger deployments, but maybe it adds some steps when the database slows down. Legacy systems often run into trouble because everything stays tightly connected and is hard to spread across servers. The new setup tries to split signaling from the data side using Flexi SIP for the actual messages and SQL tables for accounts and policies. That separation shows up in how registrations get checked and how calls get sent onward. This project address all the limitations by introducing a modular infrastructure for SIP-based communication, where Flexisip acts as a proxy, SIP signaling engine, while the external SQL systems manage subscriber and routing information.

## II. OBJECTIVES

The main goals were to cut down on internal modules and let the database handle most of the heavy lifting for users and routes. Architecture-wise, several Flexi Sip nodes are sitting behind a load balancer, along with clients and some monitoring tools. The database stores profiles and registration details so the proxy can pull what it needs without keeping everything local.

## III. SYSTEM ANALYSIS AND DESIGN

Proposed System: The proposed System architecture consists of the following components: Flexisip Server, external database, SIP clients, load balancer, monitoring and logging system, and the external DB, which consists of user profiles, domains, registration information, routing policies, device mappings, and data related to presence Flexisip communicates with the database to retrieve and manage the subscriber information dynamically.



**IV. ARCHITECTURE OVERVIEW AND DIAGRAM**

The proposed system for enterprise level VoIP architecture consists of multiple flexisip nodes connected behind a load balancer to support high availability. SIP clients communicate with flexi sip servers for registration and calling operations. The infrastructure dynamically interacts with an external database for routing, registration, call handling, and user management. The monitoring and logging systems are optional to use, but are integrated to provide the real time statistics.

**ENTERPRISE VoIP INFRASTRUCTURE ARCHITECTURE**

*Flexisip-based Scalable SIP Infrastructure with External SQL Database*

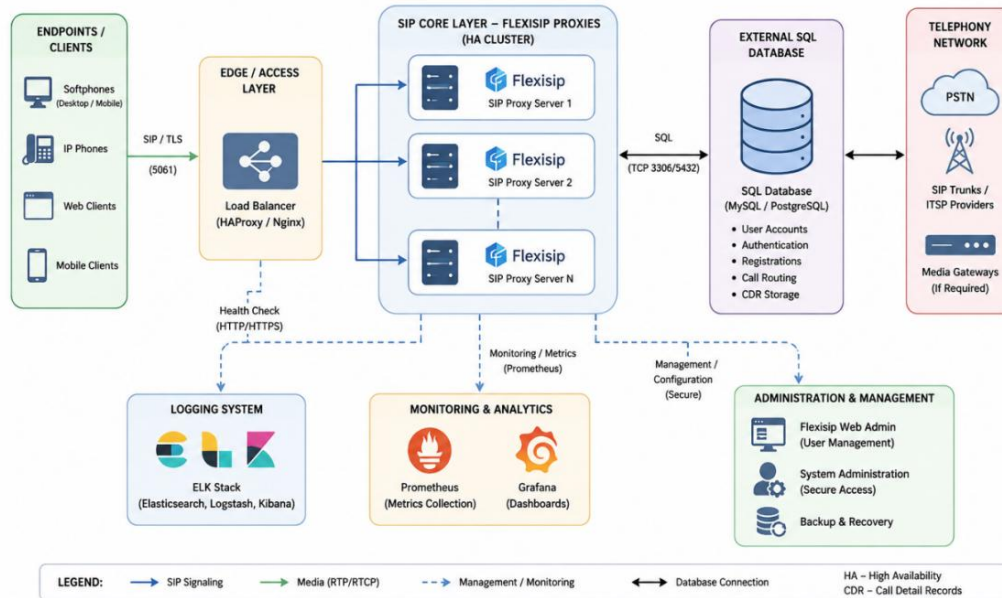


Fig:1 Architecture Overview and Diagram

**V. FUNCTIONAL WORKFLOW**

**SIP Registration:** SIP clients sends a register request to the proxy, Flexisip processes SIP headers and forwards it to the database for validation. User information is validated against the database. Currently, this system supports SQL based systems. Registration details are stored and synchronized in real time with the proxy server. **SIP Routing:** SIP routing happens when INVITE requests are received, and routing logic queries the SQL based policies that are implemented. Flexisip forwards calls to target endpoints/ subscribers. **Presence and Messaging:** Presence information gets updated dynamically with the SQL database. SQL database stores the metadata for synchronization with real time data. **Monitoring and Logging:** SIP transactions are logged, the performance metrics are collected, and the SQL queries and SIP traffic are monitored.

**VI. SYSTEM DESIGN**

**Database Design:** The SQL database contains several core tables that consist of User tables with user id, username, domain, password hash, and status. The routing table consists of route\_id, source, destination, and priority. The registration table contains reg\_id, user\_id, contact, and expires. This is an architecture of tables for simple and easy database design. **System configuration:** The Flexi sip configuration includes the SIP listener configuration, registrar module, TLS support, database connectivity, transport protocols, and IPv4 support.



### **VII. DEVELOPMENT**

The implementation of the enterprise VoIP infrastructure was carried out using the Flexisip proxy server integrated with an external SQL database to provide central user management, scalable SIP registration handling, and efficient call routing capabilities. The entire system is deployed in Linux-based environments using Debian-based systems. All the required libraries, database connectors, and development packages were installed for Flexisip Installation. The Flexisip server was configured to support all SIP signaling over UDP, TCP, and TLS transports. MySQL was used as a backend database for storing subscriber details, SIP registration information, routing policies, and authentication. Database tables were designed to manage user accounts, SIP domains, contact addresses, and registration information. SIP clients such as Zoiper and Linphone are configured using the generated SIP credentials to validate registration and call establishment functions. The system architecture also incorporates a load balancing layer using a proxy server to distribute SIP traffic across multiple Flexisip nodes for improved large-scale availability.

### **VIII. TESTING AND PERFORMANCE**

**Performance Evaluation:** Performance evaluation is done using testing parameters that include concurrent SIP registrations, parallel INVITE requests, multiple SIP endpoint connections, and querying the database under load. **Metrics Evaluated:** Using Registration latency, call setup timing, throughput, CPU utilization, memory consumption, and the database response time. These are the metrics that are used to evaluate the deployed system before launching to production or the consumer.

### **IX. MERITS AND DEMERITS**

The proposed system has advantages like high scalability, centralized management, simplified provisioning, better integration with enterprise systems, flexible routing policies, improved distributed deployment support, efficient SIP signaling performance, and easier monitoring and analysis. Each system has advantages and disadvantages. Here, the disadvantages are dependency on database availability, requirement for optimized SQL indexing, potential bottlenecks under extreme load and high transactions, and additional complexity in distributed sync. These limitations can be addressed using replication, caching, and load balancing mechanisms.

### **X. CONCLUSION**

This paper presented a scalable SIP communication infrastructure using Flexisip integrated with an external SQL database. The architecture separates SIP signaling from centralized subscriber and routing management, improving the scalability and operation. The proposed system demonstrates that Flexisip can effectively support large-scale SIP deployments when combined with SQL-based external management systems. The architecture is suitable for enterprise VoIP platforms, cloud communication systems, and distributed real-time communication infrastructures.

### **REFERENCES**

1. J. Rosenberg et al., "SIP: Session Initiation Protocol," IETF RFC 3261, 2002. [Online]. Available: <https://datatracker.ietf.org/doc/html/rfc3261>
2. Belledonne Communications, "Flexisip Documentation," Flexisip Official Documentation, 2025. [Online]. Available: <https://wiki.linphone.org/xwiki/wiki/public/view/Flexisip/>
3. Belledonne Communications, "Linphone and Flexisip Open Source Communication Stack," 2025. [Online]. Available: <https://www.linphone.org>
4. OpenSIPS Project Team, "OpenSIPS SIP Server Documentation," OpenSIPS Project, 2025. [Online]. Available: <https://opensips.org>
5. Kamailio SIP Server Team, "Kamailio SIP Server Documentation," Kamailio Project, 2025. [Online]. Available: <https://www.kamailio.org>



6. PostgreSQL Global Development Group, "PostgreSQL Documentation," PostgreSQL Official Documentation, 2025. [Online].  
Available: <https://www.postgresql.org/docs/>
7. Oracle Corporation, "MySQL Reference Manual," MySQL Documentation, 2025. [Online].  
Available: <https://dev.mysql.com/doc/>
8. A. B. Johnston, "SIP: Understanding the Session Initiation Protocol," Artech House, 2009.
9. M. Handley, H. Schulzrinne, E. Schooler, and J. Rosenberg, "SIP: Session Initiation Protocol," Request for Comments 2543, Internet Engineering Task Force, 1999.
10. Prometheus Authors, "Prometheus Monitoring System Documentation," 2025. [Online].  
Available: <https://prometheus.io/docs/>
11. Grafana Labs, "Grafana Documentation," 2025. [Online].  
Available: <https://grafana.com/docs/>
12. Elastic N.V., "Elastic Stack and ELK Documentation," 2025. [Online].  
Available: <https://www.elastic.co/docs>
13. The Linux Foundation, "Linux Networking and SIP Infrastructure Concepts," 2024. [Online].  
Available: <https://www.kernel.org/doc/>
14. ITU-T, "Overview of Recommendations for Voice over IP Systems," International Telecommunication Union, 2023. [Online].  
Available: <https://www.itu.int>
15. Redis Labs, "Redis Documentation," Redis Official Documentation, 2025. [Online].  
Available: <https://redis.io/docs/>
16. Kubernetes Authors, "Kubernetes Documentation," Cloud Native Computing Foundation, 2025. [Online].  
Available: <https://kubernetes.io/docs/>
17. D. Sisalem, J. Kuthan, and S. Ehlert, "SIP Security Challenges and Solutions," IEEE Communications Surveys & Tutorials, vol. 11, no. 4, pp. 52–63, 2009.
18. Varga and R. Hornig, "An Overview of the OMNeT++ Simulation Environment," Proceedings of the 1st International Conference on Simulation Tools and Techniques, 2008.
19. Wireshark Foundation, "Wireshark Network Protocol Analyzer," 2025. [Online].  
Available: <https://www.wireshark.org>
20. Zoiper Team, "Zoiper SIP Softphone Documentation," 2025. [Online].  
Available: <https://www.zoiper.com>

