Telecoms Services Secure based on Cloud

Abdallah Handoura
Faculty of Engineering and Technology
Muscat University, Oman

Abstract: The Next Generation Network (NGN) is an IP-based, packet-oriented telecommunications architecture designed to support a wide range of services. The IP Multimedia Subsystem (IMS), in comparison, provides the control layer and service delivery framework that enables multimedia applications to function over NGN. While NGN constitutes the core network infrastructure, IMS serves as the specialized platform responsible for delivering and managing multimedia services such as voice and video communications to end users. Security for users, services, and providers is a fundamental requirement. The growing sophistication of security threats, combined with service diversity and the widespread adoption of cloud-based and distributed systems, has made comprehensive security a primary concern. In this work, a holistic security mechanism is introduced to improve protection across the IMS environment by integrating IMS architecture with cloud computing platforms. This integration leverages the cloud's advanced security capabilities, multiple defense layers, and control mechanisms to strengthen system reliability, confidentiality, and overall robustness.

This paper presents an integration between two fields: telecommunication services and network concepts along with open technologies, aiming to provide an open service creation framework and to leverage the security mechanisms offered by cloud-based authentication.

Keywords: IMS, Security, Cloud Computing, OpenStack, SIP

1. INTRODUCTION

Next Generation Networks (NGN) can support a diverse set of services, including interactive multimedia applications such as voice and video communication, conferencing, and instant messaging, as well as non-interactive services like push-based applications, multimedia streaming, and web-based platforms such as e-commerce, e-learning, and e-health services. A major advantage of NGN lies in its ability to unify traditionally separated, vertically structured networks where data, voice, and video services are managed independently into a single, horizontally integrated network infrastructure.

The IP Multimedia Subsystem (IMS), standardized by the 3GPP as an open IP-based architecture, uses the Session Initiation Protocol (SIP) [1]for signaling and enables service convergence across heterogeneous access networks. IMS plays a central role in realizing service convergence by providing a unified service delivery framework that allows operators to offer advanced NGN services. As shown in Figure 1, the IMS architecture follows a horizontal design in which control and service layers are clearly decoupled[6].

Within the IMS framework, service providers can flexibly design their service layers to foster innovation by assembling services from reusable components. IMS supplies standardized service building blocks that can be shared among multiple application servers, facilitating service reuse and significantly shortening service development and deployment cycles[2].

Cloud computing has emerged as a leading paradigm in modern information technology, delivering scalable, flexible, and cost-effective computing resources. Cloud infrastructures provide virtualized resources over the Internet through three primary service models: Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS)[3]. These models support a wide range of applications, including multimedia services, Voice over IP (VoIP), and video streaming.

Integrating IMS with cloud computing environments offers substantial benefits by combining IMS's service delivery capabilities with the cloud's flexible resource management and security services. This integration streamlines the creation of value-added services, supports dynamic allocation of infrastructure and software resources, and enhances security for users, services, and service providers through cloud-based protection mechanisms.

In modern distributed and multi-cloud environments, ensuring secure data transmission remains one of the most pressing challenges, making security requirements paramount. The European Union Agency for Cybersecurity (ENISA) has identified key risks in cloud computing and issued guidelines and best practices addressing threats such as data leakage, malicious insiders, governance loss, and insecure data management.

Likewise, the Cloud Security Alliance (CSA) has outlined strategies for constructing robust cloud application security architectures that emphasize visibility, control, and incident remediation while encouraging adherence to security best practices. Additionally, the National Institute of Standards and Technology (NIST) has proposed several recommendations to help organizations select cloud deployment models that align with their security objectives.

Despite its advantages, securing the IMS environment remains challenging due to the diversity of services, protocols, and architectural components it encompasses. This complexity increases the potential attack surface, exposing IMS users, services, and providers to a broader range of security vulnerabilities and risks.

International Journal of Computer Applications Technology and Research
Volume 15–Issue 03, 118 – 126, 2026, ISSN: -2319–8656
DOI:10.7753/IJCATR1503.1018

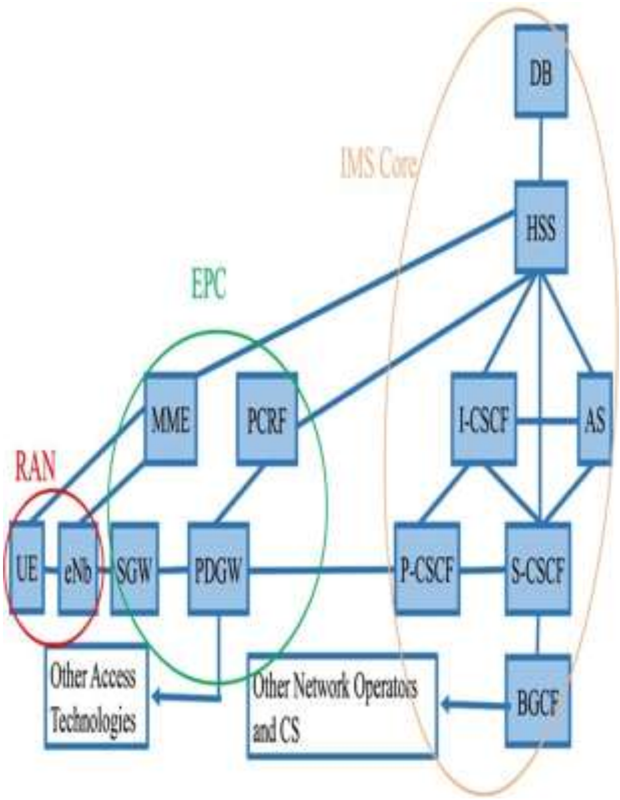

Figure 1: The IMS Architecture

1- IMS and Cloud

The fast evolution of the Internet has driven major transformations in information technology, influencing both hardware and software systems. The IP Multimedia Subsystem (IMS) is intended to operate across heterogeneous networks while enforcing policies that facilitate efficient service delivery to users. In parallel, cloud computing has become a fundamental technology by enabling the storage of massive volumes of data and offering open, multi-infrastructure platforms accessible to users.

2.1 IP Multimedia Subsystem (IMS)

The IP Multimedia Subsystem (IMS) is a core reference framework within Next Generation Networks (NGN), designed to provide a global, open service delivery platform for converged multimedia services. Its objective is to establish a standardized overlay architecture that allows users to access integrated multimedia services regardless of location, access method, or time. IMS achieves this by integrating telecommunications systems, Internet technologies, and real-time multimedia capabilities into a unified and heterogeneous environment.

Consistent with NGN architectural principles, IMS is organized into three functional layers, as shown in Figure 2: the Service Layer, the Control Layer, and the Transport Layer.

- **Service Layer (Application Layer):** Responsible for hosting, executing, and managing the services delivered to end users.
- **Control Layer:** Handles session signaling, control, and management, and coordinates traffic flow between the service and transport layers.
- **Transport Layer:** Provides core network support by interconnecting access networks with IP-based backbone networks.

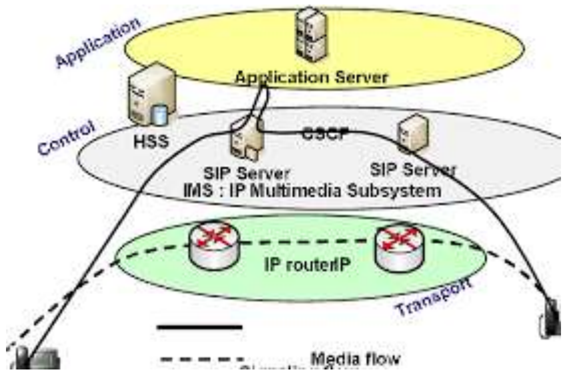

Figure 2: three layers

All telecommunications systems, regardless of their technological maturity, rely on a core set of capabilities such as session management, service control, media processing, and interconnection mechanisms to support essential communication services. In the IP Multimedia Subsystem (IMS), these functions are distributed among specialized functional entities. The responsibilities and capabilities of these entities are summarized below [4].

Call Session Control Function (CSCF)

The Call Session Control Function (CSCF) forms the central control framework of IMS, providing key functions including user authentication and authorization, session handling, message routing, and service control. A fully deployed IMS network includes three CSCF components operating together: the Serving-CSCF (S-CSCF), the Proxy-CSCF (P-CSCF), and the Interrogating-CSCF (I-CSCF).

Serving-CSCF (S-CSCF)

The Serving-CSCF is the main control entity for user sessions, services, and charging. It interacts with the Home Subscriber Server (HSS) through the Diameter protocol over the Cx interface to obtain authentication credentials and user profiles, which are used to invoke appropriate application servers.

Proxy-CSCF (P-CSCF)

The Proxy-CSCF is positioned at the access edge of the IMS network and acts as the primary entry point to the operator's domain. It is the first element to receive signaling requests from user equipment. When a user accesses the network through a visited domain, the P-CSCF forwards signaling messages to the user's designated S-CSCF in the home network.

Interrogating-CSCF (I-CSCF)

The Interrogating-CSCF is also deployed at the network edge and serves as the initial contact point for signaling messages arriving from external IMS networks. It queries the HSS to determine the appropriate Serving-CSCF and routes messages accordingly. If the HSS does not specify a particular S-CSCF, the I-CSCF selects one.

Application Server (AS)

IMS application servers provide the service logic required to deliver multimedia applications. These servers exist in several forms, including SIP Application Servers (SIP AS), IP Multimedia Service Switching Function Application Servers (IM-SSF AS), and Open Service Access–Service Capability Server Application Servers (OSA-SCS AS).

SIP Application Server (SIP AS)

The SIP Application Server is the most widely used AS type and is responsible for executing service logic for end-user applications based on SIP signaling. It supports a range of next-generation services, such as instant messaging and presence management.

Home Subscriber Server (HSS)

The Home Subscriber Server (HSS) acts as a centralized repository and enhanced Authentication, Authorization, and Accounting (AAA) entity within the IMS architecture. It stores critical information required for session and service control, including user identities, addressing information, security credentials, location data, user profiles, and service profiles. The HSS also generates security parameters necessary for authentication, integrity verification, and encryption processes.

2-2 OpenStack-Private Cloud Computing

Cloud computing enables users to access data and applications on demand from any location and at any time. It also offers several advantages, including flexible pricing models, user-friendly web-based interfaces, scalability, and independence from specific end-user devices \cite{mehr2017cloudkey}. Cloud computing services are commonly categorized into three models: Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS)[5].

Cloud platforms can be either open-source or public. Open-source cloud solutions include platforms such as OpenStack, OpenNebula, Eucalyptus, Heroku, and Cloud9 IDE, while public cloud providers include services like Amazon Web Services (AWS), Microsoft Azure, and Alibaba Cloud. In this project, OpenStack was deployed on CentOS 7. OpenStack is an open-source cloud platform that primarily supports IaaS and PaaS service models and provides a wide range of functionalities within its framework.

The primary objective of the OpenStack platform is to establish a global standard for cloud computing while offering a flexible software framework that supports the ongoing development of cloud solutions for both service providers and end users. OpenStack was initially released in October 2010 under the name Austin and consisted of only two components: Nova and Swift. By October 2025, OpenStack had evolved to its eleventh major release, expanding to include 36 modular components.

OpenStack is composed of several core components, including Horizon for the graphical user interface, Keystone for identity management and user authentication and authorization, Nova for compute services, Cinder and Swift for block and object storage, Glance for image management, Neutron for networking services based on software-defined networking (SDN), Ceilometer for monitoring, and Heat for orchestration, which enables the automated deployment of application stacks composed of multiple resources. All these services are managed and provisioned through standardized APIs with unified authentication mechanisms.

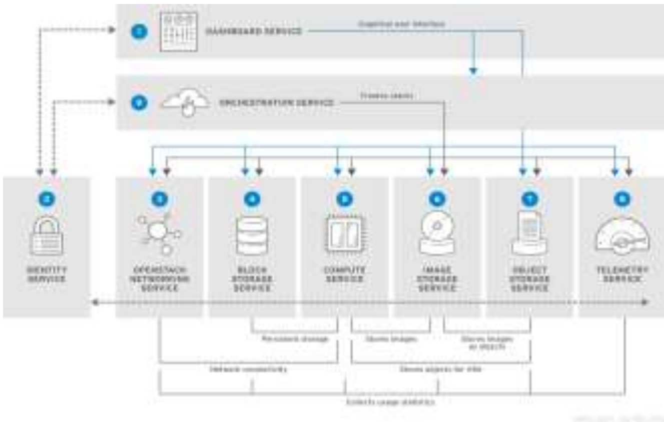

Figure 3: Openstack Components

OpenStack also includes an additional component known as the OpenStack API, which enables developers and programmers to create scripts and applications that automate the deployment and management of hardware resources through system administration and software configuration. By leveraging the OpenStack API, it is possible to automate tasks such as container deployment, service application management, web server stack configuration, elastic cloud orchestration, database operations, network traffic optimization, and platform security enforcement.

Figure 4 illustrates the OpenStack dashboard interface.

International Journal of Computer Applications Technology and Research
Volume 15–Issue 03, 118 – 126, 2026, ISSN: -2319–8656
DOI:10.7753/IJCATR1503.1018

Figure 4: openStack dashboard

OpenStack is exposed to security risks originating not only from external attackers but also from internal sources, including co-tenants and even the service provider itself. Security assessments were conducted on the server node, virtual machine instances running both Windows and Linux operating systems, and the Horizon web-based dashboard interface.

In this deployment, OpenStack was installed on a single host running CentOS 7.

Component	**Code Name**	**Description**
Compute	Nova	Provisions instances (KVM/Xen/etc.)
Image Service	Glance	Disk/server image management
Object Storage	Swift	Scalable redundant storage
Identity	Keystone	Authentication across cloud
Networking	Neutron	Network/IP management

Table 1: Openstack Compnents

3-INTEGRATION BETWEEN IMS AND CLOUD COMPUTING

The IP Multimedia Subsystem (IMS) introduces a wide range of new services and applications for users, which has significantly increased industry interest in this technology. However, the traditional IMS infrastructure faces several challenges due to the growing demand for IMS services. Conventional IMS relies on a set of SIP servers, such as CSCF entities, each performing a specific function. Additionally, the scalability of front-end distributors, which depends on expensive, specialized hardware, heavily influences SIP scalability. Consequently, traditional IMS systems often underperform compared to IT environments that leverage cloud computing, which offers exceptional scalability and availability.

Cloud computing allows for rapid and efficient deployment of new applications by pooling hardware and software resources. It also provides powerful computing capabilities and virtually unlimited storage. However, despite technical advances, different cloud providers have yet to fully meet consumer expectations. Several factors limit broader adoption:

1. **Lack of standardized signaling control:** Most cloud platforms rely on web-based interfaces to interact with clients. These interfaces cannot fully support mechanisms such as precise service access control, multiple pricing models, and other telecom-specific requirements, making it difficult for providers to deliver profitable public cloud services.
2. **Quality of Service (QoS) limitations:** Since cloud services are typically Internet-based applications, guaranteed QoS cannot always be ensured. Regardless of the type of cloud service, these systems impose significant demands on network bandwidth.
3. **Fragmented user experience:** Cloud clients face complex and platform-specific tasks, making it difficult to switch between cloud providers seamlessly, which can result in a fragmented experience for end users.

To address these challenges, the 3rd Generation Partnership Project (3GPP) introduced IMS \cite{dodd2015ims}, which has become the primary signaling architecture for Next Generation Networks (NGN) and is widely adopted by telecom operators globally. IMS's key advantage is its ability to provide standardized signaling control and configurable QoS for IP services \cite{hand2018sipsecurity}.

Despite this, IMS adoption remains limited due to the lack of innovative services. Therefore, integrating cloud computing into IMS is critical. In this architecture, cloud services act as fundamental IMS applications, while IMS provides an open, standardized service platform \cite{guise2014}. The combination of IMS and cloud computing enhances the capabilities of both technologies. User devices require capabilities such as internet access, audio/video decoding, and interactive processing via cloud-based execution, enabling rapid growth of IMS value-added services through cloud technologies.

IMS's open and standardized signaling control allows implementation of advanced service access controls, including digital rights management, charging, and security. IMS can also manage IP multimedia sessions with negotiated QoS not only at session setup but throughout the session, interacting with network components that carry application flows. Additionally, IMS can tailor features based on user profiles, locations, access networks, and devices.

Through standardized interfaces, existing IMS services—such as presence, group management, authentication, and capability negotiation—can be extended to cloud services. Cloud platforms can also leverage basic IMS services, and uniform cloud interfaces based on IMS design promote standardization across cloud computing services.

The functional architecture of cloud computing integrated with IMS is shown in Figure 5. This architecture updates IMS specifications to meet the demands of cloud services. Cloud service functions and cloud interaction functions are the two main functional categories. The IMS core handles all SIP signaling for cloud session management and service notifications, while data flows between the user equipment (UE) and the cloud platform bypass the IMS core. This architecture supports multi-provider environments.

The UE interacts with the cloud platform using multiple interfaces: the Gm interface via the IMS core for session management and service awareness, the Ut interface for managing user profiles, and the Xd interface to access cloud services. These interfaces are compatible with 3GPP IMS specifications.

User data for cloud services is divided into IMS profiles and cloud-specific profiles. IMS profiles contain information required to establish sessions and access services via application servers, while cloud-specific profiles store information needed to operate cloud services, such as enrolled service lists. IMS profile information is stored in the Home Subscriber Server (HSS), whereas cloud-specific profiles may be kept in dedicated databases, application servers, or the HSS. In systems with multiple HSS instances, the Subscription Locator Function (SLF) helps the IMS core and cloud service functions locate the correct HSS at Dh and Dx reference points, respectively.

4-SECURE SERVICES, USERS AND PROVIDERS

4.1Cloud computing Security-OpenStack

According to the *CSA Security Guidance for Critical Areas of Focus in Cloud Computing* published by the Cloud Security Alliance (CSA), the primary security controls for OpenStack focus on data protection, regulatory compliance, and operational efficiency[7]:

- **Access Management:** Implement Role-Based Access Control (RBAC) and Multi-Factor Authentication (MFA).
- **Data Encryption:** Protect data at rest and in transit using OpenStack tools, such as Cinder

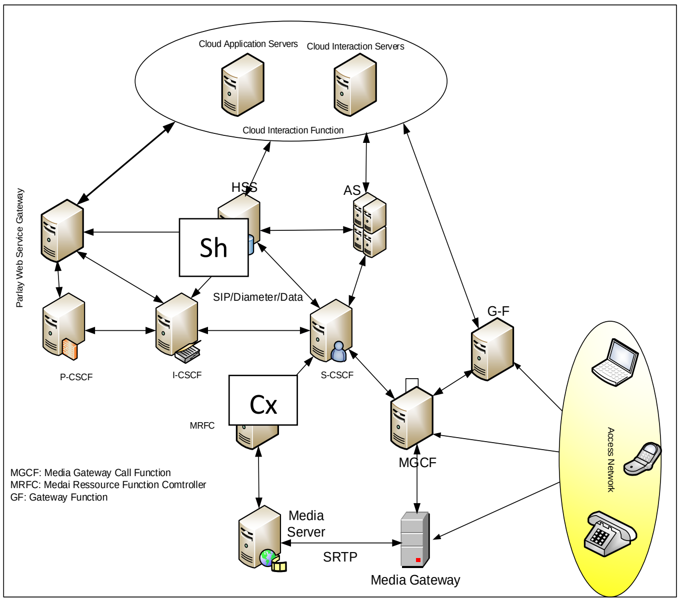

FIGURE 5: Integration between IMS and Cloud

- .
- **Network Security:** Use network segmentation and configure firewalls to minimize vulnerabilities.
- **Monitoring & Logging:** Centralize logs and enable real-time threat detection.
- **Compliance & Governance:** Adhere to standards such as GDPR, HIPAA, and PCI DSS.
- **Incident Response:** Develop response plans and regularly test disaster recovery procedures.
- **Configuration Hardening:** Secure OpenStack components, including Nova, Keystone, and Neutron.
- **Third-Party Integration Security:** Evaluate external tools and secure API interactions.

RBAC organizes user permissions based on roles rather than assigning them individually, simplifying access management and enhancing security in OpenStack environments through policy.json configuration files. When combined with MFA, RBAC provides a strong framework for secure access control.

Data encryption adds another critical layer of security in OpenStack, working alongside strict access controls to protect information both **at rest** (stored data) and **in transit** (data being transferred). OpenStack supports encryption for both scenarios through its various services, ensuring comprehensive protection of sensitive information.

Encryptions Service	OpenStack Service	Security Level
Data at Rest	Cinder	AES-256
Data at Rest	Nova	Sever-Enc
Data in transit	All Service	TLS

Table 2 : OpenStack Security Services

Network security serves as the primary defense against unauthorized access and potential breaches in private cloud environments. In OpenStack, it is a critical concern, often regarded as a top priority by cloud security professionals. Network segmentation helps create isolated zones to contain security incidents and safeguard sensitive workloads. OpenStack's Neutron networking component provides organizations with fine-grained control over their cloud infrastructure. Additionally, firewalls and security groups allow instance-level traffic management, which scales efficiently as the cloud environment expands. These tools complement network segmentation by enforcing rules directly on individual instances.

OpenStack also supports compliance with various industry-specific and data-centric regulations, such as GDPR, HIPAA, and SOX[11]. Its built-in tools streamline compliance through automated policy enforcement, making it easier to meet regulatory requirements. These tools also integrate with audit logging and monitoring systems, ensuring continuous oversight and accountability.

4-2 IMS Security

In the IMS core, securing services does not replace network security but rather focuses on protecting the data transmitted through the network and the services offered by the IMS framework[10]. This data includes information necessary for both users and services. To ensure service data security, the following measures are essential:

- **Data Privacy:** Safeguards information against unauthorized access or disclosure.
- **Service Data Integrity:** Ensures that data remains unaltered, detecting any modifications, deletions, or rerouting.
- **Service Authentication:** Divided into three types:
  - **Entity Authentication:** Confirms the identity of devices or systems during connection setup, preventing identity spoofing or impersonation.
  - **User Authentication:** Verifies the identity of human users.
  - **Data Origin Authentication:** Confirms the source of information, ensuring trust in the data origin.
- **Non-Repudiation with Proof of Origin and Delivery:** Allows recipients to verify that data came from a specific sender, and enables senders to confirm receipt by the intended recipient.
- **Access Control:** Protects network resources from unauthorized use, including reading, writing, or consuming processing and storage resources.
- **Service Anonymity:** Prevents tracking of users by operators or staff, hiding user locations and service usage while still allowing minimal necessary information for establishing connections and security purposes.
- **Security Monitoring:** Provides tools to detect, investigate, and respond to security threats.

These security mechanisms collectively ensure that IMS services maintain the confidentiality, integrity, and authenticity of both user data and service information.

4-3 Securing IMS services with OpenStack cloud

In the network architecture integrating IMS and cloud computing, IMS offers significant benefits to cloud computing, while the cloud provides multiple services to enhance IMS functionality. For our IMS–cloud integration, we utilize the open-source OpenStack platform (www.openstack.org) alongside imsOpenCore, developed by Fraunhofer FOKUS \cite{sanou2021jean}, as shown in Figure 8. OpenStack includes a central authentication and authorization component called **Keystone**, which handles authentication not only for users but also for OpenStack services. Keystone provides identity management, token issuance, service catalog, and policy services compatible with API versions 2 and 3. When a functional request is received, Keystone verifies the user's credentials (such as username, password, and URL) to confirm authorization[8]. Once verified, a token is issued containing the user's projects and associated roles. Users can then use this token for subsequent requests without re-authenticating each time. The token's expiration and validity period are configurable, as illustrated in Figure 7.

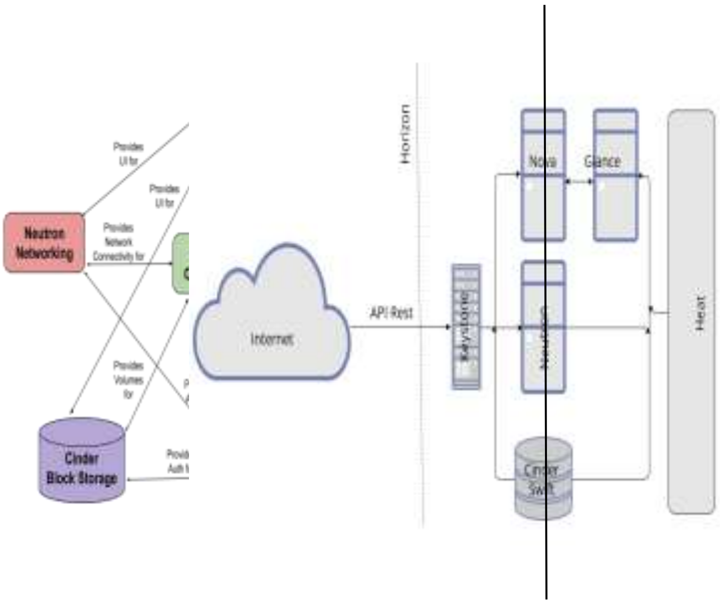

FIGURE 7: Keystone Architecture and process with different cloud components

As shown in the integration diagram in Figure 5, we set up six virtual machines (VM1 to VM6) and three containers. Each virtual machine hosts a specific software service or database (in the case of containers), dedicated to individual users[9]. External users can access these machines through a public IP address, but only after authentication and authorization are performed by Keystone services, as illustrated in Figure 8.

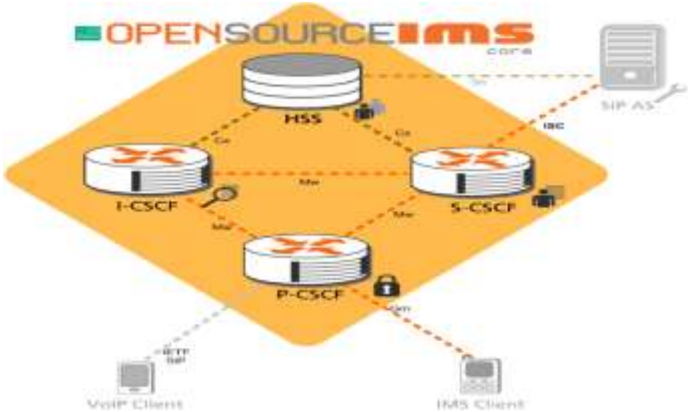

FIGURE 8: Open IMS Core

The IMS infrastructure is realized by the open source, Open IMS Core proposed by The Open IMS Core Project from FOKUS. Figure 9,

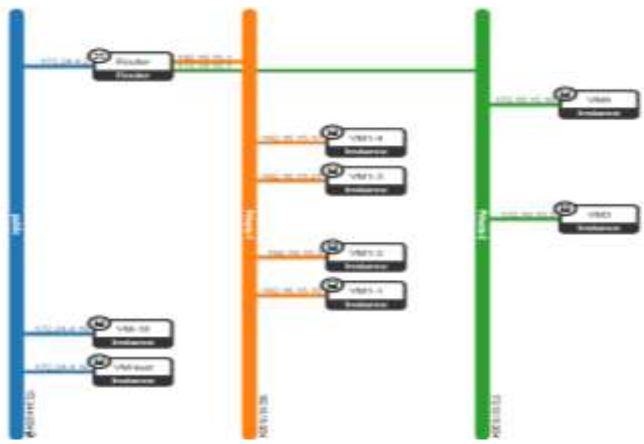

FIGURE 9: OpenStack Virtual machine topology

Create and send a REST API request to interact with an **OpenStack service (**for example, the Compute service, Nova).

1- *Step 1: Create a instance (VM-Server) via Nova*

```
POST http://<172.24.4.10/v2.1/server1
X-Auth-Token: <YOUR_TOKEN>
Content-Type: application/json
{
  "server": {
    "name": "VM1",
    "imageRef": "a6f90712-4de7-4c3c-8df4-520d7f9aab4f",
    "flavorRef": "1",
    "networks": [{ "private": "1b8b7d18-7b5f-4a94-8b47-6dfc9e2cbabc" }]
  }
}
}
```

2- *Authenticate and get a token for IMS service with Keystone*

OpenStack uses **Keystone** for identity and authentication, we must first send a **POST** request to the Keystone API to obtain an **authentication token:**

```
POST http://<172.24.4.11>:5000/v3/auth/tokens
Content-Type: application/json
{
  "auth": {
    "identity": {
      "methods": ["password"],
      "password": {
        "user": {
          "name": "admin",
          "domain": { "id": "default" },
          "password": "**********"
        }
      }
    },
    "scope": {
      "project": {
        "name": "admin",
        "domain": { "id": "default" }
      }
    }
  }
}
```

3- *Use the token to call OpenStack service*

For example, to list all servers (instances) in Nova:

```
GET http://172.24.4.15 8774/v2.1/servers/detail
X-Auth-Token: <TOKEN>
Content-Type: application/json
```

4- *Registration from an IMS user service (figure 10):*

REGISTER sips:test.example.com SIP/2.0

Via: SIP/2.0/TLS client.test.example.com:5061;
Max-Forwards: 70
From: Bob <sips:bob@test.example.com>;tag=
To: eve <sips:eve@test.example.com>
Call-ID: 12345@test.example.com
CSeq: 1 REGISTER
Expires: 0
*Contact: **
Authorization: Digest username="test", realm="test.example.com",
nonce="88df84f1cac4341aea9c8ee6cbe5a359", opaque="",
uri="sips:ss2.test.example.com",
Server: 172.24.4.1

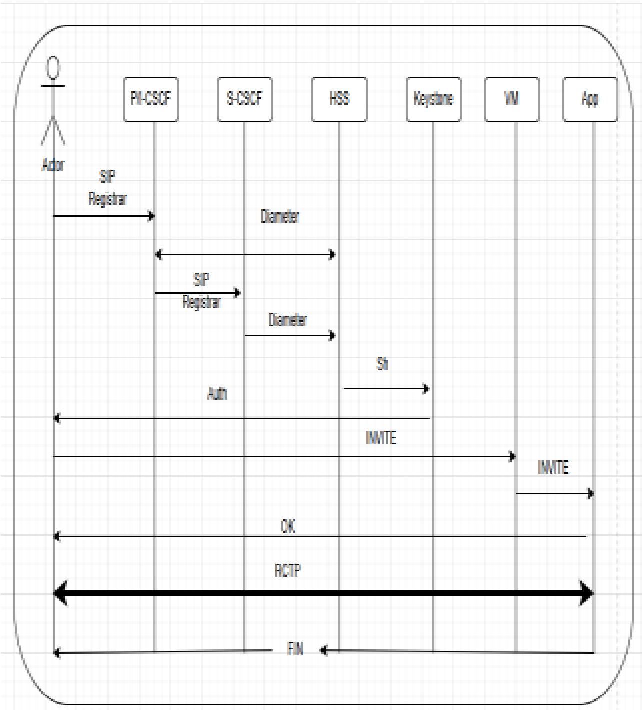

Figure 10 : Client Registrar

Another security feature provided by OpenStack is the **Security Group**. A security group (figure 11) acts as a container for a set of security rules, enabling administrators and projects to define the types of traffic and their direction (ingress or egress) that are permitted through a virtual interface port. Whenever a virtual interface port is created in OpenStack Networking, it is automatically associated with a security group. Therefore, when using OpenStack Networking, the nova.conf configuration should disable the built-in security groups and route all security group operations through the OpenStack Networking API.

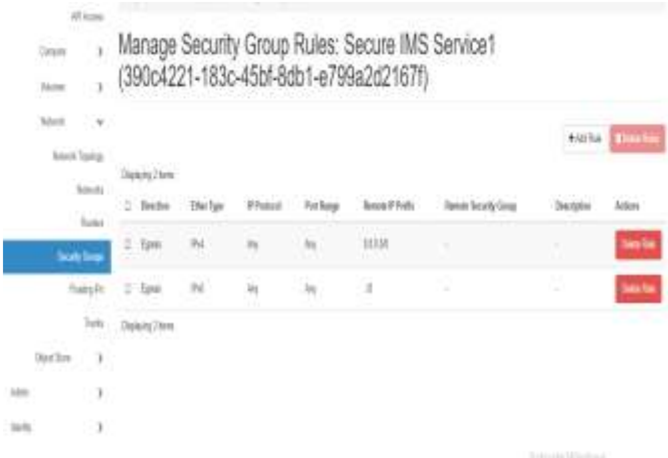

FIGURE 11: IMS Security Group in Openstack

Integrated into the IMS service and data is related to the Openstack API configured in the security group proposed by Openstack.Figure 12.

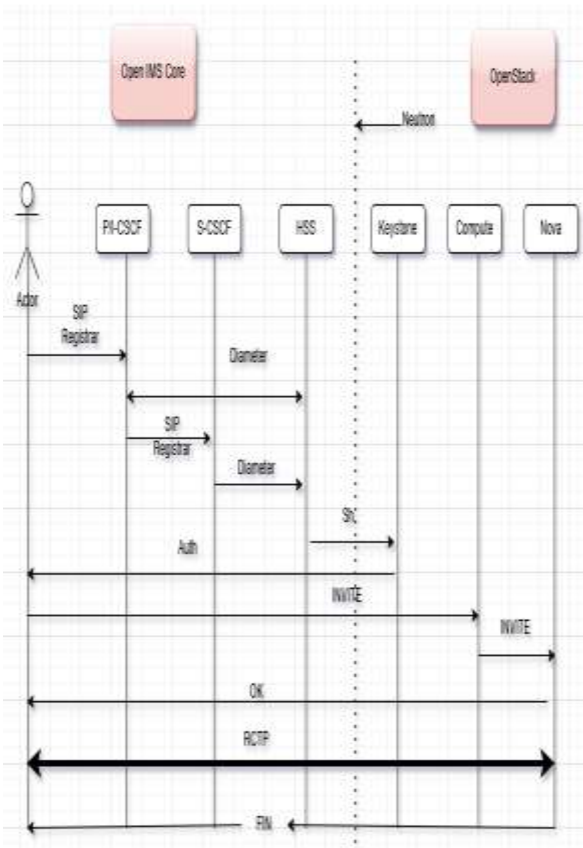

FIGURE 12: traffic Into IMS over Cloud

5- Conclusion

This paper presents a platform for integrating IMS with cloud computing using open standards. Various vendors and developers can deliver different cloud services, which are treated as standard IMS applications. IMS terminals require only minimal modifications to access these cloud services. The SIP/Diameter protocol is used to establish and manage communication between clients and cloud platforms. Once an IMS function is active, it is crucial to ensure the security of both users and services within the IMS framework.

By adopting cloud-based security mechanisms, IMS can enhance its overall security, fostering greater confidence and investment in IMS services within the industry. These security measures help users trust that their data is protected, facilitating the transition to cloud platforms and the growth of cloud service ecosystems.

Further research is necessary to tackle challenges such as scaling IMS architecture across multiple technologies and ensuring compatibility with different cloud computing providers.

REFERENCES

[1]. Dodd-Noble, S. Gundavelli, J. K. F. B. B. W, '3gpp ip multimedia subsystems (ims) option, for the internet key exchange protocol version 2 (ikev2)', rfc7651. 2015

[2]. A.Handoura, 'Secure intelligent services; sip handbook: Services, technologies, and security of session initiation protocol', Book ISBN9781315218939 Chapter 21. 2018

[3]. A.Mizani, M. A., 'Cloud-based computing; key advanced in clinical information', Elseiver pp. 239–255. 2017.

[4]. Balan, I. G. A. G. A. P. D, 'Security assessment of openstack cloud using outside and inside software tools', 14th International Conference on DEVELOPMENT AND APPLICATION SYSTEMS, Suceava, Romania, May 24-26, 2018 pp. 170–174. 2018

[5]. Donevski, S. R. M. G. A., 'Security vulnerability assessment of openstack cloud', Sixth International Conference on Computational Intelligence, Communication Systems and Networks pp. 95–100. 2014

[6]. Enisa, 'Cloud security guide fot smes', ISBN: 978-92-9204-122-9 p. https://github.com/leuk7/openIMSCoreAndServices . 2015

[7]. Luc, S. K. J, 'Openimscore school project', github p. https://github.com/leuk7/openIMSCoreAndServices . 2021

[8]. Magedanz, G. C. M. C. P. C. P. C. T. M. B. A. A. C. D. V. T, 'Cloudified ip multimedia subsystem (ims) for network function virtualization (nfv)-based architectures', IEEE Symposium on Computers and Communications (ISCC) 2014 .

[9]. Mojka.V, Mitja S, J. B. A. K. S. T, 'Ip multimedia subsystem. a guide to wireless communication engineering technologies', Book Chapter 15, 317–336. 2010

[10]. project, O, 'Openstack security guide', https://docs.openstack.org/security-guide/security group, C. (2025), 'Security guidance for critical areas of focus in cloud computing v5', CSA cloud Security Alliance . Wei Zhang. 2024

[11]. Weimin Lei. Xiao Chen, S. L, 'Architecture and key issues of ims-based cloud computing', 2013 IEEE Sixth International Conference on Cloud Computing pp. 629–635.2013