

Terms of Reference Document (ToR) Document for IOS MCN

Areas of Focus:

1. Engineering:
 - a. Software : Open-source Software Development & Delivery
 - b. Testing : Open-source Test Development & Delivery
 - c. Systems : Open-source Systems Engineering Delivery
2. Governance:
 - a. Open-source Governance Strategy, Framework and Execution

May 2024

V1.0

Contents

1.	Definition and Glossary	1
2.	Introduction	1
3.	Scope of Work	1
3.1.	Engineering	1
3.1.1.	Software: Open-source Software Development	1
3.1.1.1.	5G RAN	1
3.1.1.2.	5G SA Core	1
3.1.1.3.	IMS	1
3.1.1.4.	Platform Level Orchestration and Management	1
3.1.1.5.	Platform Level Monitoring & Observability	1
3.1.2.	Testing: Open-source Test Development	1
3.1.2.1.	Component Test Framework	1
3.1.2.2.	System Test	1
3.1.2.3.	Pre-Deployment Test Framework	1
3.1.3.	Systems: Open-source Systems Engineering	1
3.1.3.1.	CI/CD Strategy, Framework and Execution	1
3.1.3.2.	Open-source Code Management (GitHub) Framework and Execution	1
3.1.3.3.	Release Management	1
3.1.3.4.	Platform Security Strategy and Hardening	1
3.2.	Governance	1
3.2.1.	Open-source Governance Strategy, Framework and Execution	1
3.2.1.1.	Framework and Strategy:	1
3.2.1.2.	Operationalize and Maintain	1
3.2.1.3.	IOS MCN Website and Management	1
4.	Expectations from the EOI applicants	1
4.1.	Engineering	1
4.1.1.	Software: Open-source Software Development	1
4.1.2.	Testing: Open-source Test Development	1
4.1.3.	Systems: Open-source Systems Engineering	1
4.2.	Governance	1

1. Definition and Glossary

R&R : Roles and Responsibilities

CI/CD : Continuous Integration / Continuous Deployment

Eoi : Expression of Interest

CLA : Contributor Licensing Agreement (May be applicable for individual / corporates)

SBM : Software Bill of Materials

TOC : Technical Oversees Committee

TSC : Technical Steering Committee

Apache, MIT, GPL : Some of the open-source licenses

RAN : Radio Access Network

CORE : Mobile Communication Core Layer

OAM : Operation And Management for mobile communication software

NMS : Network Management System

TOR : Terms of Reference

COE : Container Orchestration Engine

IOS MCN : India Open-source for Mobile Communication Network

CRD : Custom Resource Definition

SLA : Service Level Agreement

AI : Artificial Intelligence

RBAC : Role Based Access Control

CNI : Container Network Interface

CSI : Container Storage Interface

VAT : Vulnerability Assessment Test

MFA : Multi-Factor Authentication

NFT : Non-Fungible Token

FIM : File Integrity Monitoring

SLA : Service Level Agreement

HIPS : Host-based Intrusion Prevention System

IDS : Intrusion Detection System

IPS : Intrusion Prevention System

VPN : Virtual Private Network

SSH : Secure Shell

API : Application Programming Interface

PSP : Pod Security Policies

KSPM : Kubernetes Security Posture Management

OPA : Open Policy Agent

CVE : Common Vulnerabilities and Exposures

SOAR : Security Orchestration, Automation, and Response

OS : Operating System

SAST : Static Application Security Testing

DAST : Dynamic Application Security Testing

MCS : Mobile Communication Software

UI : User Interface

IaC : Infrastructre as Code

API : Application Programming Interface

CD : Continuous Deployment

DNS: Domain Name System

2. Introduction

The client is implementing an open-source project for communication software called IOS MCN (India Open-source for Mobile Communication Network). This project will have multiple government and non-government contributors. Client wants to set this up as a global standard open-source project with right practices and methodologies.

In line with the best practices followed by globally established Open-Source Forums (Linux Foundation, Open Air Interface etc.), there is a provision to engage third parties to procure software components as well as testing, engineering and open-source governance services to help ruggedise and strengthen the code base and deliveries.

Given the mandate of the IOS-5GN project to provide in a time bound manner, commercially deployable deliveries and support, it is essential that such best practices of engaging the vendors and experts from the ecosystem are adopted. The following 2 step process will be followed by IOS-5GN:

1. STEP 1 – Qualifying vendors for participation in the RFP process for any components or services needed by IOS-5GN. This TOR for EOI is intended for this qualifying process.
2. STEP 2- On identification of specific components or services for which third parties need to be engaged – creation of an RFP to invite the vendors qualified in STEP 1 to provide Commercial bids.

It should be noted that:

1. Not all the items listed in the TOR for EOI will result in RFPs.
2. The pre-qualification of vendors through the EOI process is needed to expedite the RFP process.

Typically, this project will be based on existing open-source projects along with new code developed and added.

The contributors and partners will be at different geolocations. The project will be implemented with contributors from different locations and affiliated organizations. The governance and engineering practices should be with global open-source standards and best practices based.

The client wants to make it a flagship open-source project from India and there may be global open-source projects and foundations collaborations during the project's development.

The project quality should be high to ensure faster commercial deployment of this project by vendors. This is one of the top priorities, and hence, the project execution and strategies should be aligned to ensure this result.

This is a call for expression of interest to contribute to this project by contributing to the project in the areas detailed below. There are multiple areas of work expected which need different set of competency and industry experience. Currently, for the following broad focus areas, the expression of interest (EOI) are invited. You can apply for any broad areas or specific smaller areas of work.

1. Engineering:

- a. Software : Open-source Software Development & Delivery

b. Testing : Open-source Test Development & Delivery

c. Systems : Open-source Systems Engineering Delivery

2. Governance:

a. Open-source Governance Strategy, Framework and Execution

The project has strict timelines, and the vendors/service providers should abide by the same.

3. Scope of Work

3.1. Engineering

3.1.1. Software: Open-source Software Development

3.1.1.1. 5G RAN

- a. Support O-RAN based DU & CU with a combination Radio Units for both Low Power (LP) and High Power (HP) deployments.
- b. Support for the following O-RAN components:
 - near-RT RIC - Enables near-real-time control and optimization of O-RAN elements
 - non-RT RIC - Enables non-real-time control and optimization of O-RAN elements and resources
 - CU-CP - Hosting the RRC and the control plane part of the PDCP protocol
 - CU-UP - Hosting the user plane part of the PDCP protocol and the SDAP protocol
 - DU - Hosting RLC/MAC/High-PHY layers based on a lower layer functional split
 - RU - Hosting Low-PHY layer and RF processing based on a lower layer functional split
 - xApps - Independent Software plug-in
- c. Support for the following O-RAN interfaces:
 - Support of F1-c interface between DU to CU-CP for Control Plane traffic
 - Support of F1-u interface between DU to CU-UP for User Plane traffic
 - Support of E1 interface between CU-CP and CU-UP
 - Support of E2 interface between near-RT RIC and CU (CU-CP and CU-UP)
 - Support of E2 interface between near-RT RIC and DU
 - Support of Open FH S-Plane to handle time, frequency, and synchronization between the DU and RU
 - Support of Open FH CU-Plane for Control & User-plane communication between DU and RU
 - Support for Open FH M-Plane interface between DU and RU
 - Support for Open FH M-Plane interface between non-RT RIC and RU
 - Support N2 for Routing of Control Plane traffic between CU-CP and AMF
 - Support N3 for Routing of User Plane traffic between CU-UP and UPF
- d. Support for Xn and N2 Handover
- e. Support of n78 frequency band. With option to support other frequency band.
- f. Support for 2T2R and 4T4R MIMO Configuration
- g. Support for Channel Bandwidth of 40MHz, 100MHz
- h. Support for 3GPP Release 15 or 16
- i. IEEE 1588 Synchronization

- j. Support for Network slicing with appropriate 5QI values
- k. Support for MBB, URLLC and MMTC Use cases
- l. Should support VoNR & ViNR
- m. Should support open APIs for integration to any O&M (OSS/BSS/SMO/RIC integration), MEC platform integration, OT/IT integrations.
- n. Support Functions for Radio Resource Management: Radio Bearer Control, Radio Admission Control, Connection Mobility Control, Dynamic allocation of resources to UEs in both uplink and downlink (scheduling)
- o. Support N2 and N3 interfaces with the 5G Core. Routing of User Plane traffic towards UPF(s) using N3 and Routing of Control Plane traffic towards AMF using N2 interface.
- p. Peak data rate of 650 Mbps in DL and 230 Mbps in UL
- q. Each sector should support 64 active subscribers
- r. IPV4 and IPV6 should be supported

3.1.1.2. 5G SA Core

- a. Support for 5G SA Core Network Function as per 3GPP Release 15 or 16
 - Access and Mobility Management Function (AMF)
 - Session Management Function (SMF)
 - Policy Control Function (PCF)
 - Network Repository Function (NRF)
 - Authentication Server Function (AUSF)
 - Unified Data Management (UDM)
 - Unified Data Repository (UDR)
 - User Plane Function (UPF)
 - Network Slice Selection Function (NSSF)
- b. Support WiFi/non-3GPP integration using Non-3GPP Inter-Working Function (N3IWF)
- c. Support non-3GPP application function integration using Network Exposure Function (NEF)
- d. Support for Network analytics using Network Data Analytics Function (NWDAF)
- e. Support for urLLC, mMTC and TSN Features
- f. 5G Core shall support centralized and distributed packet core architectures
- g. Solution shall support interworking with any third party 3GPP Rel15/16 compliant O-RAN or gNodeB
- h. Shall support multiple simultaneous Active Bearer per UE and 5QI values for VoNR, ViNR
- i. Shall support Handover over NGAP/N2 interface
- j. 5G Core shall support Network functions virtualization (NFV)
- k. Support Network segmentation/slicing for custom QoS or data connectivity
- l. Support separate User Plane (UP) functions from the Control Plane (CP) functions, allowing independent scalability, evolution and flexible deployments e.g. centralized location or distributed (remote) location
- m. Should support open APIs for integration to any O&M (OSS/BSS/SMO integration), MEC platform integration, OT/IT integrations.
- n. 5G Core Network utilises service-based architecture (SBA) that spans across 5G functions and interactions including authentication, security, session management and aggregation of traffic from end devices
- o. Support for IPv4 and IPv6 stack support
- p. UDM support 10K subscriber configuration

- q. Active subscribers of 2K subscribers with 20K active PDU sessions
- r. Peak aggregated UPF throughput of 10 Gbps
- s. Run each 5G Core Network Function as container
- t. Run on Intel or AMD or ARM platform

3.1.1.3. IMS

- a. **Support for N5 support in 5G Core for signalling between IMS and PCF**
- b. **Support for SBI capable Open-source IMS entity for VoNR**
 - Support for HTTP 2.0 with security for SBI between IMS and 5G Core
 - N5 interface for Rx to PCF. Support for AF Capability
 - N70 interface for I-CSCF and S-CSCF
- c. **Support N6 for SIP and RTP exchange between IMS and UPF**
 - Use of 5QI=5 (non-GBR) for SIP signalling messages
 - Use of 5QI=1 (GBR) for speech packets after connection is established.
 - Use of 5QI=2 (GBR) for video packets after connection is established.
 - Support for dynamic gBR bearer creation based on SIP message exchange
- d. **Support for voice and video calls:**
 - Should support minimum 100 simultaneous voice calls.
 - Should support minimum of 10 simultaneous video calls.
 - Support voice conference calls of 8 subscribers or more
 - Support video conference calls of 4 subscribers or more
- e. **Special Capabilities**
 - Support for VoNR and GBR QOS capability in RAN
 - Support for VoNR and multiple bearer QOS profile configuration capability per Subscriber using the SMO
 - Support for IMS DNN configuration using SMO
 - Support for per subscriber voice/video telemetry statistics collection and its observability
- f. **Platform Requirement**
 - The VoNR IMS should run as a container
 - Run on Intel or AMD or ARM platform
 - VoNR IMS and 5G Core can be installed in different hardware and communicate using IPv4 or IPv6 address
 - Support 1K concurrent VoNR sessions
 - Support 200 concurrent ViNR sessions

3.1.1.4. Platform Level Orchestration and Management

The project is being developed to provide complete mobile communication software with various components to realize RAN, CORE and OAM. The key intent of the project is to

provide a complete solution which will have a commercial level of quality so that the vendors can build products based on this with reduced time to market.

The overall project deployment, orchestration, operation and management from the project level are critical for successfully using this solution. This is a horizontal-level feature which can help for the easy deployment of the solution onto a new environment and operation management of this environment. Please note that this is not the core feature of the mobile communication software solution; rather, this helps to orchestrate the components from outside the core features. The client wants the service provider to build the project-level orchestration, operation and management as a separate software component and can facilitate the actual mobile communication software deployment.

The project deployment is expected to be as cloud native way with containerized workloads.

The orchestration operation management module needs to be a separate sub-project developed to enable the deployment and management of IOS MCN software in different use cases. It must be scalable and maintainable.

High level view of the scope you can refer to the below picture:

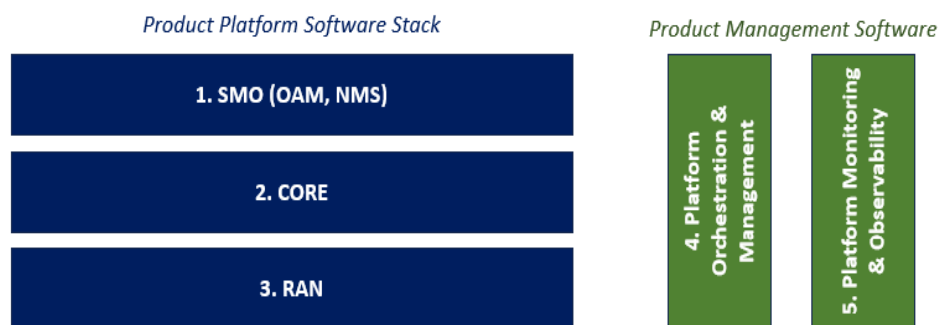


Figure 1 : IOS MCN High Level Software Stack

1 to 3 are the real communication software solution modules. 4 provides a software module which can help to deploy and do project level operation and management for the complete solution on a given environment. This is to ensure the solution is correctly deployed and the health of each components deployed are tracked. So, 1, 2 and 3 are the business application or target application which will be orchestrated and managed by 4 – Platform Orchestration & Management. This section of software development scope is for that.

The mobile communication software components will be mostly containerized (may be with some components as independent processes). It is expected to develop the Platform Orchestration & Management software based on any container orchestration engine (COE) like Kubernetes.

The rest of this document will be using “mobile communication software” (MCS) as a common term to refer the components of 1, 2 and 3.

Some of the specific requirements are given below:

- a. **Automated Orchestration**
 - Able to provide an automated orchestration of MCS
 - Supports image repository integrations for auto download of MCS components
 - Supports COE like Kubernetes or container native orchestration
 - Able to orchestrate both containerized and non-containerized MCS modules
 - Able to orchestrate MCS as distributed deployment as needed

- b. **Deployment Management**
 - Support hybrid deployment (on premise, cloud or across)
 - Provide deployment progress, error reporting and logs
 - Can provide programmatic deployment (deployment software can use it automatically through a program)
 - Ability to debug the issues quickly
 - Provide UI and command line-based orchestration

- c. **Discovery & Scalability**
 - Provide scalability of MCS components horizontally on demand
 - Support for scaling both stateless and stateful applications
 - Orchestration level support for the availability of MCS components through methods like replication or similar
 - MCS services recovery (dynamic and services can communicate with each other)
 - Support for service registries and DNS-based service discovery OR similar methods to manage the service discovery

- d. **Load balancing**
 - Service level (on top of MCS components) load balancing (evenly distribute the load for specific MCS components) based on the resource utilization.
 - Support the integration with external load balancers for traffic ingress and egress

- e. **Versioning, Auth and Configurations**
 - Supports the rollback and versioning of MCS components
 - Secured key or auth methods for MCS components as needed
 - Provide APIs / plugins/configuration methods for extending or customizing the MCS orchestration
 - Support Infrastructure as code (IaC) methods and principles for defining and provisioning MCS components deployment infrastructure

- f. **Visualization**
 - Appropriate UI based visualization support for orchestration and management
 - Appropriate Command line-based support for orchestration and management

- g. **COE (Container Orchestration Engine)**
 - Supports COE like Kubernetes or container native O&M
 - Provide compatible solution for easy maintenance
 - Support custom resources through methods like CRD(Custom Resource Definition) in Kubernetes

- h. **Configuration**

- Provides simple and flexible configuration management
- Storage of project level configurations separate from the MCS components level configurations
- Configurations versioning supported (track changes and history)
- Configuration updates without application downtime (unless there is a hard constraint from MCS components)

i. Reporting

- Provides reporting feature to give reports on MCS software level health and various resource-level information (or similar)
- Scheduling and configuration of reporting feature
- Basic monitoring with the listing of services, clusters, nodes, and resources
- Provides access to the system operation logs
- Support basic alerts (minimum - resource utilization thresholds, MCS components crash)

j. Resource Management & Backup

- Supports the overall resource management for MCS components: allocation and management of compute resources – CPU, Memory and Storage)
- Configurations and policies for resource quotas and limits to prevent resource contention and ensure fair allocation.
- Replication & Replica Management for MCS components
- Backup and backup management of MCS components
- Disaster recovery for MCS components

k. Integrations

- Supports integration of any project level monitoring solution which can provide extended monitoring/observability capabilities (Prometheus, Grafana, or Kubernetes-native monitoring tools or similar or more)
- Support integration of backup/restore solutions (COE native solutions like SODA Kahu, VMWare Velero for Kubernetes Backup/Restore etc)
- Support the integration of extended security solutions
- Easy integration of continuous deployment (CD) tools and solution

l. Documentation & Resource Utilization Reports

- Provide complete documentation
- Provide the resource utilization of MCS orchestration components

3.1.1.5. Platform Level Monitoring & Observability

The project is being developed to provide a complete mobile communication software which will have various software components to realize RAN, CORE and OAM. The key intent of the project is to provide a complete solution which will have a commercial level of quality so that the vendors can build products based on this with reduced time to market.

The overall project-level infrastructure monitoring for running various IOS MCN application components is critical for successfully using this solution. This is a horizontal-level feature which can help with the project-level monitoring and observability of the solution for complete IOS MCN. Please note that this is not part of the core features of mobile communication software solutions; rather, this helps to monitor/observe the complete IOS MCN from outside for its resources, software health, and performance and provide the right alerts at the right time.

The client wants the service provider to build the project-level monitoring and observability software as a separate software component and facilitate the actual mobile communication software health and resource management. The projects components may be containerized and the monitoring software needs to support cloud native or hybrid deployment models.

The solution needs to be scalable and maintainable.

Please refer the Figure 1 : IOS MCN High Level Software Stack

1 to 3 are the real communication software solution modules. 4 provides a software module which can help to monitor and provide comprehensive observability for the complete solution on a given environment. This is to ensure the health and availability of the IOS MCN solution. So, 1, 2 and 3 are the business application or target application which will be monitored or provide observability for, through 5 – Platform Monitoring & Observability. This section of software development scope is for that.

The mobile communication software components will be mostly containerized (maybe with some components as independent processes). It is expected to develop the Platform Monitoring / Observability (project level) software based on any container orchestration engine (COE) like Kubernetes Or can integrate as a native solution to such COEs.

The rest of this document will be using “mobile communication software” (MCS) as a common term to refer the components of 1, 2 and 3.

Some of the specific requirements are as follows:

a. Automated & Container Native

- Supports COE like Kubernetes or container native monitoring/observability
- Support native integration with COEs (like Kubernetes, Docker Swarm or Apache Mesos or similar as agreed with IOS MCN. Kubernetes is preferred for capability showcasing)
- Able to provide an automated monitoring and observability solution of MCS
- API interfaces for monitoring and observability

b. Comprehensive Monitoring

- Provides Metrics for container health, resource usage (CPU, Memory, Storage and Network), and any custom resources
- Metrics collection from containers, orchestrators (e.g., Kubernetes), and underlying infrastructure
- Able to monitor both containerized and non-containerized MCS modules
- Monitor the interaction between microservices within containerized modules of IOS MCN
- Policy based monitoring

c. Logging and Log Management

- Centralized logging which provides log classification, access, versioning and rotation as needed
- Ability to Capture logs from standard output/error streams, application logs, and system logs
- Provide log consolidation/aggregation, searching, filtering, and visualization

d. Integrations

- Supports standard telemetry interfaces like Open Telemetry or similar
- Supports integration of new tools to extend monitoring or observability features
- Supports distributed monitoring integrations
- Provide custom data management for the monitoring data as agreed with IOS MCN
- Need to support external or internal storage endpoints

e. Alert Management

- Supports standard telemetry interfaces like Open Telemetry or similar
- Support Alert Management System providing alerts and notifications for the predefined thresholds and behaviour anomalies
- Alert management through SMS, Emails, Chat or any incident management tools (based on the decision from IOS MCN)
- Policy based alerting

f. Scalable

- Additional instrumentation support for new metrics or alert collection
- Plugin and exporter model for extendibility
- Scalable to handle large volume of data and large deployment scenarios (the SLA to be decided mutually)
- Minimum resource utilization for monitoring/observability solution so as to provide minimum overhead running the monitoring agents or components
- Efficient query and filtering mechanisms for large metrics data
- Mechanism to manage the storage and policy-based lifecycle management for monitoring data

g. Secure

- Ensure the security for the communication between monitoring components and IOS MCN containers.
- RBAC-like access management for monitoring features and data
- Ensure encryption of sensitive data at rest and in transit

h. Artificial Intelligence

- Basic AI integration for anomaly detection, trends and forecasting (the selection models, accuracy and expectations to be agreed upon with IOS MCN during the development)

i. Visualization

- Provides good visualization with customization options (preferably based on templates) which can provide metrics, logs and alerts on a real-time manner with appropriate history levels.

j. Documentation & Resource Utilization Reports

- Provide complete documentation
- Provide the resource utilization of MCS orchestration components

3.1.2. Testing: Open-source Test Development

a. Automation and integration

- Identification and integration of Unit Test (UT) Frameworks
- Development of Unit Level Testing Framework (eg., Jmeter, Junit, Cunit, GoTest) for the 5G Core, RAN and SMO
- Support to integrate all tools and tests with CI/CD Pipeline
- Ensure sufficient code coverage and reporting

b. Test Design and Testing

- UT test design for 5G Core, RAN and SMO
- Add new test cases to the Unit Testing Framework
- Modify/delete test cases from the Unit Testing Framework
- Creation of Documentation and FAQ for the Unit Test Frameworks

3.1.2.1. Component Test Framework

a. Automation and integration

- Development and Integration of Selenium, Appium or any Automated Testing Frameworks for 5G Core, RAN and SMO
- Automation and Reporting
- Automate the generation of the resiliency events and load for capacity and performance test.
- Deploy Citrus like tools (API Level Testing and Microservice Integration Testing)
- Support to integrate all tools and tests with CI/CD Pipeline
- Ensure sufficient code coverage and reporting

b. Test Design and Testing

- Complete Test Design for components testing

- Compliance testing of 5G Core, RAN and SMO using 3rd Party tool or Open-source Frameworks
- Test the 5G Core Network Functions using a 3rd Party Testing Tool by simulating UE and RAN
- Test the CU, DU using a 3rd Party Testing Tool by simulating UE, RU and 5G Core
- Test the SMO using 3rd party tool (eg. Cypress Comprehensive UI Testing) its integration into CI pipeline for Frontend Testing.
- Add new test cases to the Component Open-source Test Framework or 3rd party tools
- Modify/delete test cases from the Component Open-source Test Framework or 3rd party tools
- Creation of Documentation and FAQ for the Unit Test Frameworks

3.1.2.2. System Test

a. Strategy

- Overall System strategy for IOS MCN will be a mix of agile and iterative test methodologies.
- Define functional end-to-end and conformance/interface-level tests.
- Define longevity tests to verify the stability of the SUT/DUT.
- Identify tests and validation points to evaluate the resilient nature of the SUT/DUT.
- Determine the frequency and expected execution time of automated test suites based on the test phase.
- Define the test phases and the various test suites that will make up the test phases.
- Finalize the acceptable failure threshold for each test phase.
- Determine the exit and entry criteria for every test phase.
- Baseline the capacity and the performance criteria.
- Define the call model distribution for performance tests.
- Identify requirements related to testbed and human resources.

b. Automation and integration

- Integration of Selenium, Appium for automation of SMO configuration and monitoring
- Integration of Robotium (BlackBox Testing for Android) for automation of UE clicks
- Integrate 3rd Party Testing Tool or Source Test Framework for end-to-end automation of 5G Core, RAN and SMO testing
- Support to integrate all tools and tests with CI/CD Pipeline
- Ensure sufficient code coverage and reporting

c. Test Design and Testing

- Complete test design for system testing
- Add new test cases to the System Test Framework
- Modify/delete test cases from the System Test Framework
- Test the scalability and performance using 3rd Party tools.
- Creation of Documentation and FAQ for the Unit Test Frameworks

d. Support and Maintenance

- Defects reporting and management
- Support and maintenance for the system test artefacts

- Field trials

3.1.2.3. Pre-Deployment Test Framework

- a. **Ability to comprehend the Customer requirement for Private 5G and its use-cases**
 - Understand the customer requirement of 5G in detail from an end-to-end solution perspective
 - Creation of a Private 5G Solution Architecture based on Customer Requirement
 - Deployment of a Private 5G pre-deployment test setup for testing the customer use cases
- b. **Ability to integration of the use cases (Fleet Management, Video surveillance, Robotics, Drone Survey, IoT Monitoring, Mission Critical Communication etc.)**
 - Integrate the 3rd party applications into the Private 5G pre-deployment test setup in private Cloud or connectivity to Public Cloud
 - Integrate the 3rd party User Equipment to the Private 5G pre-deployment test setup and test the end-to-end use-case
 - Collaborate with all the stake-holders – 5G Core, RAN, SMO team and 3rd party HW vendor and application provider to get the end-to-end use-case working
- c. **Ability to do Acceptance Testing as per customer requirement:**
 - Manual testing of all the use cases required by customer
 - Scalability and performance testing of those use cases as per feasibility and availability of HW
 - Creation of the Acceptance test report comprising the test cases, its result, any work-around etc.
- d. **Automation of Pre-Deployment Test Setup**
 - Regression testing of the Pre-Deployment Test Setup when new patch fixes or new images are available and reporting the test results
 - Ability to automate the Pre-Deployment test-cases to the extent feasible and its integration to the CI/CD pipeline

3.1.3. Systems: Open-source Systems Engineering

3.1.3.1. CI/CD Strategy, Framework and Execution

- a. **Design Strategy, Framework and Deployment**
 - Design the CI/CD Strategy for IOS MCN Open-source
 - Identify the right framework, tools and libraries
 - End to end deployment and maintenance of CI/CD pipeline for IOS MCN
 - Hybrid CI/CD support covering On-Cloud (eg. GitHub Actions) and On-Premise (eg. Jenkins) Build and Deployment Strategy.
 - Support for the integration of variety of Testing Frameworks on Unit, Integration, System, Acceptance, A/B testing for languages like C, C++, Go, Java, Python and more.
 - Deployment strategies like Canary, Blue-Green Ocean Strategy
- b. **Special CI/CD Features**

- Multi Team Collaborative feature support and RBAC enabled user management controls.
 - Multiple pipeline chaining mechanism and pipeline queuing mechanism.
 - Integration of Tools (T-shark) and Plugin (Docker, Maven, Prometheus, Git) - as needed for the pipeline
 - Support for code maintainability, version control and management.
 - Scale the CI/CD pipelines to meet up with the required demands.
 - Multi-site CI/CD Infrastructure from different Geo-Locations with minimal latency.
 - User-end-support and Documentation for proper operation of the CI/CD Infrastructure.
 - High Availability requirements.
- c. Training and Support**
- Basic training to use and work with the CI/CD framework for all the internal teams
 - Provide technical support.

3.1.3.2. Open-source Code Management (GitHub) Framework and Execution

- a. Open-source Code Management Strategy, Guidelines and Templates**
- Comprehensive open-source code management strategy covering all components of IOS MCN
 - Development Guidelines for all Sub projects
 - Repositories organizations strategy and deployment
 - Prepare templates and management (PR, Issue, Project ReadMe, package management etc)
 - GitHub (or similar) Branching Strategy
 - GitHub (or similar) Security Strategy
- b. Creation of Organizations, Repositories and Management**
- Creation, Configurations and Management of all Organizations for Sub Projects (example – GitHub orgs)
 - Creation, Configurations and Management of all repositories across all organizations (private/public)
 - Downstream the base code for each sub projects
 - Security and Secret Management
 - Access Management and Member Management
 - Deploy and maintain branching strategy
- c. Workflows, Hooks and CI/CD Integration**
- Support for work flows and hooks (like integrations, pipelines, GitHub actions etc)
 - Support for the integration with CI/CD external tools
 - Create custom workflows: Use GitHub Actions to create custom workflows for automating various tasks within the organization
 - Manage env variables and settings
- d. PR, Issues and Projects**
- PR triggers, rules and management

- Issues guidelines, labels and linking to projects and management
 - GitHub (or similar) projects creation, templates, support and management
 - Setting up teams and management
 - Project creation, examples and templates
 - Support to manage projects (fields, reporting management)
- e. Audit, Monitoring and Maintenance**
- Regular audit for all organizations and repositories
 - Regular monitoring for all organizations and repositories
 - Regular maintenance for all organizations and repositories
 - Audit Reporting
 - Regular Backup or mirroring
- f. Release and hygiene files Management**
- Tags and release management support
 - Basic hygiene files management (like maintainers, readme, install, .GitHub, and more)
 - Release automation and trigger support
- g. Base-code and Upstream**
- Base-code sync-up as needed
 - Implement upstream/downstream strategy through rules, automation, branching, syncing and rebasing
- h. Training and Support**
- Provide basic training for the usage and contributions for all internal teams
 - Provide technical support

3.1.3.3. Release Management

- a. Deploy Release Strategy and Maintain**
- Deployment of release strategy across all release repositories
 - Create, Manage and maintain all releases across all repositories
 - Master IOS MCN release management linking all the dependent sub projects and modules
 - Deploy version management and versioning across the project and sub projects
- b. Creation, Automation and artifacts**
- Creation and management of all releases (tags, versions)
 - Releases asset management
 - Packaging Support
 - Integration support for CI/CD
 - Release notes (audit, consistency and manage)
- c. Training, Support and Documentation**
- Release management training
 - Release creation and management support
 - Comprehensive documentation for release management

3.1.3.4. Platform Security Strategy and Hardening

a. Security Strategy

- Comprehensive Security Strategy for IOS MCN project and all sub projects
- Guidelines and templates
- Industry best practices for open-source security assessment and hardening

b. VAT

- Ability to do VAT (Vulnerability Assessment Test) and reporting for different software components of IOS MCN (application, libs and dependencies)
- Ability to do Open-source VAT exploiting the available open-source vulnerability data based on the components used or imported for IOS MCN
- Regular scanning and detect vulnerability and reporting
- Vulnerability data base update

c. Scanning and Incident Reporting

- Project and Sub Projects Security Scanning
- Facility for scheduled and event driven scanning and reporting
- Incident reporting through SMS, Email, Chats or any incident management systems
- File Integrity Monitoring (FIM) for all changes (files, directories and configs)
- Quick Alerts (SLA mutually decided)
- Incident response strategy, plan and protocols

d. Assessment and Hardening

- Configuration interface for assessment and hardening
- Support pre build, build, pre-release and post release security assessment and hardening
- Facility to patch IOS MCN with security updates without downtime (unless there is a constraint from IOS MCN side)
- Secure the configuration, deployment and monitoring solutions running on top of IOS MCN

e. Access Management

- Support overall management of Access control and privilege with strong auth methods/MFA
- Optional security support through NFT or similar solutions
- Support VPN / SSH or similar for remote access and management of the system

f. Protection

- Endpoint protection from malware, virus, ransomware, or similar malicious threats
- Host-based intrusion prevention systems (HIPS) for endpoints
- Develop IDS/IPS
- Protect from basic network-based threats

g. Container Security – Policy, Strategy and Hardening

- Specific container security strategy
- Policy based COE security hardening (like Kubernetes cluster security hardening etc)

- PSP (Pod Security Policies) for COE along with measures to handle the pod security contexts, access, privileges, host namespaces, volume mounts and configurations
- Ensure Container network policies and secure network communication between pods based on namespace, labels, and network policies

h. Container Security - Ensure

- Secure COE API server and related components (example Kubernetes API Server, control manager, scheduler, etcd datastore etc)
- Provide container image security (with tools and methods)
- Protocol and strategy for security management of dependencies, APIs etc for IOS MCN. The same will be followed by IOS MCN software components
- Secure storage for sensitive data like API keys, Auth, tokens, password and other keys
- Run time security scanning, logging, reporting and alerting (for the COE)

i. Container Security – Access and Practices

- Utilize COE level security practices and methods like KSPM
- RBAC to manage the access to COE resources (like Kubernetes resources access)
- Specific granular level of access definitions based on roles, accounts or applications

j. Integration, Training and Documentation

- Leverage COE security tools from its software ecosystem (like Falco, OPA/Gatekeeper, and kube-bench for Kubernetes)
- Extendable options for security assessment and hardening for the COE (for example, adding new APIs natively in Kubernetes through the operator framework)
- Ability to integrate standard security assessment and hardening tools
- Consolidated APIs for visualization and integration with IOS MCN solution
- Provide complete documentation
- Provide the resource utilization of security assessment and hardening components
- Training and Support

[Remaining part of the page is left blank]

3.2. Governance

3.2.1. Open-source Governance Strategy, Framework and Execution

This comprises the complete open-source project governance, licensing strategy and release strategy for the project. It delivers the strategies and its deployment in the project.

Assess the current project state and gaps, then refine as needed. The overall governance structure, memberships, high-level operation practices, open-source ecosystem, and basic governance operations, and building release and licensing strategies imbibing global open-source best practices.

The governance must include the organization structure (including R&R), operation practices, membership models, execution and maintenance. The team must join specific committees to drive the activities and ensure the success of the project operation.

Overall project level licensing and release strategies to enable compliant and efficient project release. This should be scalable and maintainable.

3.2.1.1. Framework and Strategy:

This comprises generating, maintaining and mobilising the approval process for the artefacts that define the items listed below on the website or repository. It further entails operationalising the related governance processes and enabling sustainable institutional mechanisms for the same.

a. Charter, Framework and Models

- Open-source project / foundation governance framework creation
 - Governance structure refining to suit the project
 - Charter preparation and roadmap
 - Rules, Regulations and Working Procedures detailing:
 - Roles and Responsibilities
 - Governance Artifacts Templates
- Support in creation and maintenance of artifacts (like Charter, R&R, Working Procedures etc.) including Policies and Guidelines
- Imbibe best practices from global open-source models

b. Technology Management and Strategy

- Technology Group Management Strategy and guidelines
- Technical decisions and operation strategy
- Technical Ladder coordination and operation strategy (Sub Project management)

c. Membership, Outreach and Communication Strategy

- Membership Strategy, Model and Guidance
- Open-source Communication Strategy (internal)
- Outreach (external) to build project communication and awareness
- Community Building Strategy

d. Licensing, Development and Release Strategy

- Comprehensive Licensing Strategy for the project
- Development Strategy (GitHub and more)
- Upstream – Downstream Strategy
- Release Strategy

e. Assessment, Refining and Training

- Regular assessment of the governance and reporting
- Refine and correct as needed
- Bring best practices from open-source and industry
- Training and awareness

3.2.1.2. Operationalize and Maintain

a. Governance

- Prepare and maintain all the artifacts
- Build the project governance based on the strategy (work with all groups and stakeholders to deploy it)
- Clear Roles and Responsibilities documented and imparted
- The communication methods are to be documented
- Clear governance level meetings management guidelines and support to practice
- Institutionalise the review and approval process of governance artefacts
- Support for tracking methods and deploy
- Appointment management
- Join and support top governance committees and drive the execution

b. Membership

- Deploy the membership models
- Complete process and guidelines to manage members
- Membership Management artefacts to be provided (onboarding, managing, agreements)
- Members management – flow and maintenance

c. Technology Management

- Technology management model deployed and on boarded
- Coordinate and support TOC (Technical Oversees Committee or similar)
- Open-source project or sub projects or modules incubation and its strategy (time to time)
- Project Operation Inputs (Project reporting/management strategy, project and sub-project organization, org structure and responsibility refinement)
- Open-source and industry analysis support for technology areas for the project

d. Licensing

- Deploy end to end licensing for all the sub projects, repositories and releases
- Ensure the licensing strategy alignment
- Inputs to improve the licensing strategy time to time
- CCLA and ICLA preparation and support to deploy and manage
- Complete license database for all the projects, sub projects and modules

- Assessment and licensing compatibility
 - Analysis and deployment of right tools for licensing and related SBM (Software Bill of Materials)
 - LICENSE and other related files templates and ensure deployment
 - Oversee the use of any Black Duck or other software provenance analysis tools
- e. Development**
- GitHub (or similar) strategy deployment for the project and sub projects
 - Deployment Guidelines and templates (contributors, process and more)
 - Ensure for upstream-downstream strategy & expert support
 - Technical Artifacts (Requirements, Design, Architecture) Templates
 - Sub-project establishment and monitoring
 - Open-source Project Evaluation templates and methods
 - Development tools recommendations and management
 - Establish suitable support and consulting structure for code releases
 - Code quality methods and recommendations
- f. Release**
- Deploy Release Strategy
 - Prepare release templates and support in maintenance
 - Expert Support in open-source release management across project and sub project (regular)
 - Release process quality assessment
 - Release checklist (for project and sub projects)
 - Release tools and methods recommendations

3.2.1.3. IOS MCN Website and Management

- a. Design and Layout**
- Website design and layout
 - Sitemap
 - Theme
 - Align to project and sub project artifacts (logos, colour or related)
 - Responsive design
- b. Contents**
- Content preparation
 - Align with Project roadmap, Stakeholders and finalization
 - Review and maintenance of the contents
- c. Menu, Sub Menu, Pages, Forms development and deployment**
- Menu and Sub menus
 - All pages as per the sitemaps and contents provided
 - Forms (Contact, Members and more)
 - Ensure the responsiveness
 - Maintenance support

d. Website Management and Maintenance

- Uptime and reporting
- Security
- Backup / Restore
- User and backend management
- Notifications
- Data and Media management
- Hosting management
- Sub Domain management (if any)

[Remaining page is left blank]

4. Expectations from the EOI applicants

- The applicants are expected to have prior experience in incubating/building open-source projects/foundations in the specified areas.
- They need to furnish the details of their competency and credibility to fulfil the requirements of each activity.
- The applicant can provide EOI for one or more areas specified in this document
- Some of the specific expectations are added for each area separately below. That can be a reference set of competencies and applicants can add all other relevant information to substantiate their applications.

4.1. Engineering

4.1.1. Software: Open-source Software Development

a. Expertise in 5G Development

- Working experience of developing or contributing in 5G RAN
 - Knowledge of CU protocol - RRC, SDAP and PDCP
 - Knowledge of DU - RLC/MAC/High-PHY layers
 - Knowledge of near-RT RIC or non-RT RIC
 - Knowledge of open source 5G RAN Stack like OAI, SRSRAN
- Working experience of developing or contributing in 5G Core
 - Knowledge of Network Function – SMF, AMF, AUSF, PCF, UPF, NSSF, UDM, UDR
 - Knowledge of Service Based Architecture and SBI protocol
 - Knowledge of RAN Integration using N2, N3 and IP Routing using N6
 - Knowledge of open source 5G Core like SD-Core, Free5GC, OAI
- Working experience of developing or contributing in IMS
 - Expertise with SIP Signalling and SDP parameters
 - Experience with RTP and RTCP packet format
 - Understanding of P-CSCF, I-CSCF, S-SCSF
 - Working knowledge with Open-source IMS (eg. Kamailio)
 - Knowledge of N5 interface and 5G Application Functions (AF)

- Working Experience with User Plane Performance Optimization and making Scalable Systems Open-source contribution experience and efficiency with open-source tool usage
- Programming and Containerization
 - Experience with containerization of components and Kubernetes
 - Proficiency with C, Java, Golang, Python languages
 - Experience in the development of features for IMS , 5G Core and 5G RAN
 - Experience in development of high-performance systems with knowledge of Multi-threading, multi-processor
 - Experience with Data Plane Development Kit (DPDK), Vector Packet Processing (VPP)

b. Expertise in Platform Level Orchestration and Management

- Experience and expertise in container based application development and orchestration
- Expertise in any one COE like Kubernetes (Kubernetes preferred)
- Experience building and deploying applications on any COE
- Developed new cloud native components within COE (like Kubernetes native components within COE to manage applications)
- Expertise in COE APIs, Operators and features (need to show case example projects developed)
- Experience building Orchestration and O&M solution for project with multiple components
- Expertise in automated deployment of applications
- Expertise in automated installation methods like helm, ansible (or similar)

c. Platform Level Monitoring & Observability

- Experience and expertise in container based application development and orchestration
- Expertise in any one COE like Kubernetes (Kubernetes preferred)
- Expertise in COE APIs, Operators and features (need to show case example projects developed)
- Experience building monitoring/observability project for infrastructure management
- Expertise in application monitoring
- Experience in OpenAPI, Open Telemetry etc (standards in containerized application domain)
- Kubernetes API level work experience, especially for runtime, CNI or CSI
- Expertise in building any monitoring or observability products or open-source projects with customer trials
- Capability in infrastructure monitoring

- Monitoring projects with GUI/Visualization, Exporters and plugins
- Capability in Kubernetes monitoring and related native tools like Prometheus and Grafana

4.1.2. Testing: Open-source Test Development

a. Expertise in UT

- Expertise and competency in integration of Unit Testing Framework (eg., Jmeter, Junit, Cunit, GoTest) for the 5G Core, RAN and SMO into the CI pipeline
- Working knowledge of 5G Core Network Functions, 5G RAN components and Management Functions
- Experience and proficiency with scripts for adding new test cases and maintaining to the Unit Test Framework
- Experience in writing test cases based on the relevant 3GPP technical specification documents.
- Experience in preparing test data, defect logging, test execution, and test result analysis

b. Expertise in Component Test

- Expertise and competency in integration of Selenium, Appium Testing Frameworks for 5G Core, RAN and SMO into the CI Pipeline
- Experience in Core or RAN or SMO testing for 3GPP Release 15 or Release 16
- Working knowledge of 3GPP Compliance testing of 5G Core, RAN or SMO
- Experience with 3rd party testing tools or open-source framework for 5G Core Network Functions
- Experience with 3rd party testing tools or open-source framework for 5G CU, DU
- Experience with 3rd party testing tools or open-source framework SMO
- Experience in Citrus (API Level Testing and Microservice Integration Testing) into the CI pipeline.
- Experience and proficiency with scripts for adding new test cases and maintaining to the Component Test Framework
- Experience in writing test cases based on the relevant 3GPP technical specification documents.
- Experience in preparing test data, defect logging, test execution, and test result analysis.

c. Expertise in System Test

- Experience of Selenium, Appium for automation of SMO configuration and monitoring

- Experience of Robotium (BlackBox Testing for Android) for automation of UE clicks
- Experience in Core or RAN or SMO testing for 3GPP Release 15 or Release 16
- Working knowledge of end-to-end system testing with 5G Core, RAN and SMO
- Experience with 3rd Party Testing Tool or Source Test Framework for end-to-end automation of 5G Core, RAN and SMO testing
- Experience and proficiency with scripts for adding new test cases and maintaining to the System Test Framework
- Experience with scalability and performance testing using 3rd Party tools.
- Experience in writing test cases based on the relevant 3GPP technical specification documents.
- Experience in preparing test data, defect logging, test execution, and test result analysis.

d. Expertise in Pre-Deployment Test

- Experience working with customer to understand customer requirement for 5G use cases
- Experience of creation of 5G Solution Architecture for the use cases and its deployment
- Experience of deployment of Private 5G with multiple customers for various use cases
- Working knowledge and experience of integrating 3rd party use cases to a Private 5G Network (Fleet Management, Video surveillance, Robotics, Drone Survey, IoT Monitoring, Mission Critical Communication etc.)
- Experience of setting the connectivity within Private Network and its reachability to Internet.
- Experience of configuration of the 5G SIMs, 5G Core, RAN, Application Servers to setup the Private 5G Network and Private Cloud and its monitoring.
- Ability to optimize or find work-around to ensure customer requirement for Private 5G is deployable with existing feature or resources

e. Expertise in Execution Strategy

- Pre-build Phase (Source Code): Expertise in Integration of all unit level checking frameworks into the Continuous Integration pipeline for automated code checking once pipeline is triggered.
- Post-build Phase (Binary): Expertise in Functionality testing like APIs between Network Functions once code is successfully built into binary.

- Post-Containerization (Docker Image): Expertise in integration level testing like microservice reachability and network exposures.
- System Testing (Manual Testing or semi-automated along with Targeted Device): Expertise in End to End testing of the full Stack along with Core, RAN, UE Device for validating proper functionality of code and services.

f. Expertise in Device Automation

- Experience in Android, iOS device simulation with capability of simulation and mimicking user inputs using automated scripts. (eg. ADB, Termux)
- Knowledge of Android and iOS Operating Systems and security protocols.
- Experience with in log capturing and debugging from remote devices wirelessly over wireless ADB, Putty and Log Cat

4.1.3. Systems: Open-source Systems Engineering

a. Expertise in CI/CD Strategy, Framework and Execution

- Expertise in handling CI/CD Pipeline for multiple repositories and multiple branches in single pipeline.
- Skill-set in containerization of applications, orchestration and networking (eg. Kubernetes, Container Engine and Registry, Replica-set and Load-balancing).
- Multi Team Collaborative feature support and RBAC enabled user management controls.
- Experience with Hybrid CI/CD support covering On-Cloud (eg. GitHub Actions) and On-Premise (eg. Jenkins) Build and Deployment Strategy.
- Experience in integration of Tools (T-shark) and Plugin (Docker, Maven, Prometheus, Git).
- Ability to work with opensource software life-cycle and project structure.
- Experience with handling multi-site CI/CD Infrastructure from different Geo-Locations with minimal latency.
- Experience in integration of variety of Testing Frameworks on Unit, Integration, System, Acceptance, A/B testing for languages like C, C++, Go, Java, Python and more.

b. Expertise in Open-source Code Management (GitHub) Framework and Execution

- Experience incubating new open-source projects
- Experience leading complete engineering of open-source project or foundation projects
- Experience bootstrapping open-source projects and new releases from scratch (GitHub or similar)
- Experience maintaining multiple source code repositories for open-source projects (GitHub or similar)
- Experience working with upstream and downstream projects and its execution

c. Expertise in Release Management

- Experience leading open-source projects, releases and customer trials
- Experience leading open-source releases for multiple projects / foundation (GitHub or similar)
- Experience playing key roles in open-source projects / foundations and done good collaboration and understanding of open-source practices
- Experience preparing release strategy for multi-project open-source releases

d. Expertise in Platform Security Strategy and Hardening

- Experience and expertise in container based application development and orchestration
- Expertise in any one COE like Kubernetes (Kubernetes preferred)
- Experience building and deploying applications on any COE
- Developed new cloud-native components within COE (like Kubernetes native components within COE to manage applications)
- Expertise in COE APIs, Operators and features (need to showcase example projects developed)
- Experience building security solutions for any COE (preferrable – Kubernetes)
- Expertise working in hybrid deployments of applications

4.2. Governance

d. Expertise in Open-source Governance

- Experience in incubating new open-source projects
- Experience in incubating new open-source foundations or playing key roles in the same
- Experience playing key roles in open-source projects/foundations such as advisory roles, TSC / TOC, Workgroups, Community Management
- Experience leading complete engineering of open-source project or foundation projects
- Experience in playing key roles in open-source projects/foundations such as advisory roles, TSC / TOC, Workgroups, Community Management
- Experience in Open-source governance structure preparation, charter preparation/review, membership agreement preparation/review and preparing governance artifacts
-

e. Expertise in Open-source Membership

- Experience in incubating new open-source projects / foundations with multiple members or partners globally
- Experience in Open-source membership models and strategy incubation
- Experience working on membership management, agreements and onboarding for open-source projects/foundations
- Experience working / building community around open-source projects

f. Expertise in Open-source Licensing

- Experience in incubating new open-source projects and managing the licenses
- Experience in incubating new open-source foundations and contributions to the licensing
- Experience in licensing consultancy, international talks/sessions or papers
- Understanding of prominent open-source licenses like Apache, MIT, GPL etc
- Experience working multi license projects
- Experience leading open-source projects, releases and customer trials
- Experience building licensing strategy for open-source project(s)

g. Expertise in Open-source Development

- Experience incubating new open-source projects
- Experience leading complete engineering of open-source project or foundation projects
- Experience bootstrapping open-source projects and new releases from scratch (GitHub or similar)
- Experience maintaining multiple source code repositories for open-source projects (GitHub or similar)
- Expertise in upstream and downstream management

h. Expertise in Open-source Releases

- Experience leading open-source projects, releases and customer trials
- Experience leading open-source releases for multiple projects / foundation (GitHub or similar)
- Experience playing key roles in open-source projects / foundations and done good collaboration and understanding of open-source practices
- Experience preparing release strategy for multi-project open-source releases

i. Expertise in Website Management

- Experience Managing Open-source Project / Foundation websites
- Experience developing contents and maintaining open-source project / foundation websites
- General website design, development and maintenance experience

[Remaining Page is left blank]

[End of the Document]