

Dr. Chandra R. Murthy
Dept. of Electrical Communication
Engineering Indian Institute of Science,
Bangalore – 560 012 Tel. 080-2293-2464
Email: cmurthy@iisc.ac.in



Request for Quotation

1. Project Overview

As part of the MeitY-funded India Open Source for 5G Networks project, we are initiating procurement to establish lab infrastructure. We invite qualified vendors to submit quotations to set up an Open-Source Framework to enable efficient development and delivery of high quality IOS-MCN releases in a timely manner. This will include the development strategy, implementation of open-source frameworks and selection, deployment and management of appropriate tools, adherence to appropriate licensing agreements, and comprehensive management of execution and release processes.

2. RFQ Details

RFQ Number: **FSID/IOSMCN/14-01**
RFQ Issue Date: 09th Oct 2024
RFQ Deadline: 23rd Oct 2024
Delivery Date: 30th Oct 2024
Delivery Location: IISc, Bangalore
Contact Person: Chethan (+91 6363672928)

3. Scope of Work:

The RFQ is focused on establishing, operationalizing and maintaining the Open-Source Development Infrastructure to ensure timely and high-quality releases of the IOS-MCN Open-Source Software platform. The framework should be based on globally established standards and best practices for Open-Source Code Releases. This should include CI/CD, Open source code management platform (e.g. GitHub), Release management, Licence Management, Containerisation etc. The inputs for technical evaluation of the vendors should cover the following.

1: Open-source Software Engineering:

A: 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-Premises (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 Support

- Basic training to use and work with the CI/CD framework for all the internal teams.
- Provide technical support.

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 workflows 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. Pull Request, 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: 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

4. Licensing, Development and Release Strategy

- Comprehensive Licensing Strategy for the project
 - 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
- Development Strategy (GitHub and more)
 - 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
- Upstream – Downstream Strategy
 - Upstreaming with open-source projects and platforms
 - Synchronization of releases
 - Backward Compatibility
 - Enable the development, deployment and monitoring of upstream and downstream strategy
- Release Strategy
 - 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

D. 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

If the vendor has not responded to the TOR, <https://ios-mcn.org/> please provide your input/response to the TOR also. This is a mandatory input for the technical evaluation.

4: Technical Requirements and Milestones:

Phase	Deliverables	Timeline
Phase 1:	<p>1: Implement a hybrid CI/CD pipeline using GitHub Actions for cloud-based builds and Jenkins for on-premises support, enabling smooth integration for SMO with team collaboration</p> <p>2: Deploy automated CI pipelines for Core and RAN components, integrating unit, integration, and system testing frameworks for multiple languages</p> <p>3: Implement advanced deployment strategies (Canary, Blue-Green) to ensure reliable rollouts of Core components, with multi-site CI/CD infrastructure ensuring scalability and low-latency.</p> <p>4: Integrate key tools such as Docker, Maven, TShark, and Prometheus, with pipeline queuing and chaining mechanisms to support complex workflows for both SMO and Core/RAN builds.</p> <p>5: Provide training for internal teams on using the CI/CD pipelines, with full documentation and ongoing technical support to ensure proper maintenance and high availability.</p>	Months 1-2
Phase 2:	<p>1: Establish GitHub organizations and repositories for Core, RAN, and SMO projects with security, branching strategies, and secret management, ensuring downstream base code sync and repository organization.</p> <p>2: Implement automated CI workflows for RAN and SMO components using GitHub Actions, integrating static analysis tools and supporting CI/CD integration with external tools.</p> <p>3: Define development guidelines, PR triggers, issue management, release automation, and create templates (PR, issues, README) for efficient code and project management.</p> <p>4: Implement regular audits, monitoring, and backup strategies for all repositories, with reporting on code quality, security, and performance.</p> <p>5: Provide training on using GitHub, workflows, and contributing to the open-source codebase, along with ongoing technical support for internal teams.</p>	Months 3-4

Phase 3:	<p>1: Implement and manage the release strategy for all repositories, linking the master IOS MCN release with dependent sub-projects and modules.</p> <p>2: Deploy and maintain version control across all repositories and sub-projects, ensuring consistent versioning practices.</p> <p>3: Automate the creation, management, and packaging of releases (tags, versions), integrating with CI/CD pipelines to ensure seamless deployment.</p> <p>4: Streamlining of Roles and access</p> <p>5: Manage release notes for consistency and audit purposes, along with proper asset management and packaging support.</p> <p>6: Ensure ongoing maintenance of all release processes, supporting updates, and monitoring release consistency across the project.</p>	Months 5-7
Phase 4:	<p>1: Deploy and manage licensing across all repositories and sub-projects, ensuring alignment with project goals and handling CCLA/ICLA processes.</p> <p>2: Implement GitHub strategies with contributor guidelines, upstream-downstream synchronization, and development tools management, ensuring code quality and structure.</p> <p>3: Establish and monitor upstream/downstream processes, ensuring backward compatibility and smooth synchronization of project releases.</p> <p>4: Deploy a robust release strategy with templates, quality assessments, and tool recommendations, ensuring consistent and high-quality releases across all sub-projects.</p> <p>5: Conduct regular governance assessments, implement improvements, and provide training to ensure adherence to best practices in open-source and industry standards.</p>	Months 8-12

5. Submission Requirements:

The quotation should be submitted in the form of 2 separate documents as specified below:

1. TECHNICAL EVALUATION DOCUMENT: This should contain the following:

- a. A detailed workplan based on the technical requirements based on the section (Please check any deviation from requirements).
- b. Detailed specifications of any tools that will be required for the execution of the work plan.
- c. Detailed description of additional resources (including human resources) that will be needed for the execution of the work plan.
- d. Specify any dependencies, known risks and mitigation plans.
- e. If the vendor has not responded to the TOR, <https://ios-mcn.org/> please provide your input/response to the TOR also. This is a mandatory input for the technical evaluation.

2. COMMERCIAL BID DOCUMENT:

- a. Quotations should include detailed pricing, including unit prices, taxes,, and any additional fees.
- b. Provide information on warranty terms, technical support services, and maintenance agreements.
- c. Include company profile, relevant experience, and references from past projects.
- d. Quotes for the product must be enclosed in a password protected PDF file format.
- e. Quotations must be addressed to

**Director,
Foundation for Science Innovation and Development
Innovation Centre, IISc Campus Near Maramma Circle gate
Bengaluru 560012 GSTIN: 29AAECF1802E1Z1**

3. Evaluation Criteria:

The evaluation criteria include for considering the quote and awarding with the purchase order includes,

- Compliance with RFQ requirements and specifications.
- Price competitiveness.
- Vendor experience, and track record.
- Warranty and support offerings.

- Technical capabilities and compatibility with existing infrastructure.

4. Important Notes:

- The lowest-priced quotation may not necessarily be selected; quality, reliability, and vendor reputation will also be considered.
- IOS MCN reserves the right to reject any or all quotations and to award the purchase order on its own evaluation criteria.
- Any clarifications or questions regarding this RFQ should be directed to the contact person listed above.