User Requirements Specification

# Project Overview

The aim of the project is to deploy a micro-service implementing a Java/Groovy gNMI/gRPC client to install, manipulate, and delete configuration of network devices and to view the operational data. Streaming subscription must be considered as part of the required goals.

# Mandatory User Requirements

The following user requirements have been identified for the micro-service:

1. The microservice must have two interfaces:
2. northbound interface: responsible for receiving requests and managing subscriptions. It defines the input and output APIs for the microservice using Open-API, and includes a field for identifying the intended network element for each request.

RPCs: Set, Get, Subscribe (stream, poll, once), and Capabilities.

1. southbound interface: forwards or translates requests to their intended nodes, using a unique identifier provided by the northbound interface to ensure proper delivery.

RPCs: Set, Get, Subscribe (stream, poll, once), and Capabilities.

1. The micro-service must allow users to install, manipulate, and delete configuration of network devices.
2. The micro-service must allow users to view operational data from the network devices.
3. The micro-service must support streaming subscriptions for operational data.
4. The microservice should be stateless, scalable, and performant. It should be designed for easy replication and deployment across large-scale networks and high-volume data processing environments. Its replicas should not share information with each other.
5. The micro-service must be tested thoroughly, including unit testing, integration testing, and regression testing.

# Optional User Requirements

1. The micro-service must be secure, including authentication and authorization for users and access control for network devices.
2. The micro-service must run on a cloud-based platform, such as AWS or Google Cloud Platform.

# Mandatory Constraints

The following constraints have been identified for the project:

1. The micro-service must be implemented using Java/Groovy and the gNMI/gRPC framework.
2. The microservice must be developed using the Spring Boot framework.
3. The micro-service must be documented clearly and completely, including technical documentation, user manuals, and API specifications.