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 has to be considered as part of the required goals.

# User Requirements

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

1. The micro-service must support the following RPCs:
   * Set
   * Subscribe (stream, poll, once)
   * Get
   * Capabilities
2. The micro-service must be implemented in Java/Groovy using the gNMI/gRPC framework.
3. The micro-service must allow users to install, manipulate, and delete configuration of network devices.
4. The micro-service must allow users to view operational data from the network devices.
5. The micro-service must support streaming subscriptions for operational data.
6. The micro-service must define input and output APIs for the micro-service using Open-API.
7. The micro-service must be secure, including authentication and authorization for users and access control for network devices.
8. The micro-service must be scalable and performant, supporting large-scale networks and high-volume data processing.
9. The micro-service must be tested thoroughly, including unit testing, integration testing, and regression testing.
10. The micro-service must be documented clearly and completely, including technical documentation, user manuals, and API specifications.

# Acceptance Criteria

The following acceptance criteria have been identified for the micro-service:

1. The micro-service must successfully support the specified RPCs.
2. The micro-service must be implemented in Java/Groovy using the gNMI/gRPC framework.
3. The micro-service must successfully allow users to install, manipulate, and delete configuration of network devices.
4. The micro-service must successfully allow users to view operational data from the network devices.
5. The micro-service must successfully support streaming subscriptions for operational data.
6. The micro-service input and output APIs must be defined using Open-API.
7. The micro-service must be secure, including authentication and authorization for users and access control for network devices.
8. The micro-service must be scalable and performant, supporting large-scale networks and high-volume data processing.
9. The micro-service must be thoroughly tested, including unit testing, integration testing, and regression testing.
10. The micro-service must be documented clearly and completely, including technical documentation, user manuals, and API specifications.

# 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 micro-service must run on a cloud-based platform, such as AWS or Google Cloud Platform.
3. The micro-service must be secure and comply with any relevant security standards or regulations.
4. The micro-service must be scalable and performant, supporting large-scale networks and high-volume data processing.
5. The micro-service must be documented clearly and completely, including technical documentation, user manuals, and API specifications.