Software Requirements Specification

for

Netanol

Version 1.4

Prepared by DATA DELVERS

NTS Netzwerk Telekom Service AG

07.01.2024

Table of Content

Revision History

Name	Date	Reason For Changes	Version
Michal Malovecky	3/10/2023	Document Creation	1.0
Tristan Losada Benini	8/10/2023	Document revision & alteration	1.1
Denes Takacs	8/10/2023	Document revision	1.2
Michal Malovecky	6/1/2024	Document revision	1.3
Tristan Losada Benini	6/1/2024	Document revision & minor changes	1.4

1. Introduction

1.1 Purpose

The principal goal of Netanol is to gather, process, and present Flow data produced by network devices. Its ultimate objective is to augment transparency within a network, thereby affording prospects for enhanced load optimization and more efficient troubleshooting. Through proficient management and analysis of the data, Netanol enables a more profound comprehension of network performance, empowering system administrators to make informed decisions and optimize resource allocation.

1.2 Document Conventions

- **Netanol** is the project name.
- **Project Wiki** is to be found on GitHub and provides all necessary technical documentation.

1.3 Intended Audience and Reading Suggestions

This document is intended for network administrators or project managers seeking to acquire comprehensive knowledge regarding the technical aspects and prerequisites of the Netanol project. Its purpose is to provide detailed information and insights into the project's technical framework and necessary conditions. By utilizing this document, network administrators and project managers will be equipped with the essential understanding required to effectively utilize Netanol.

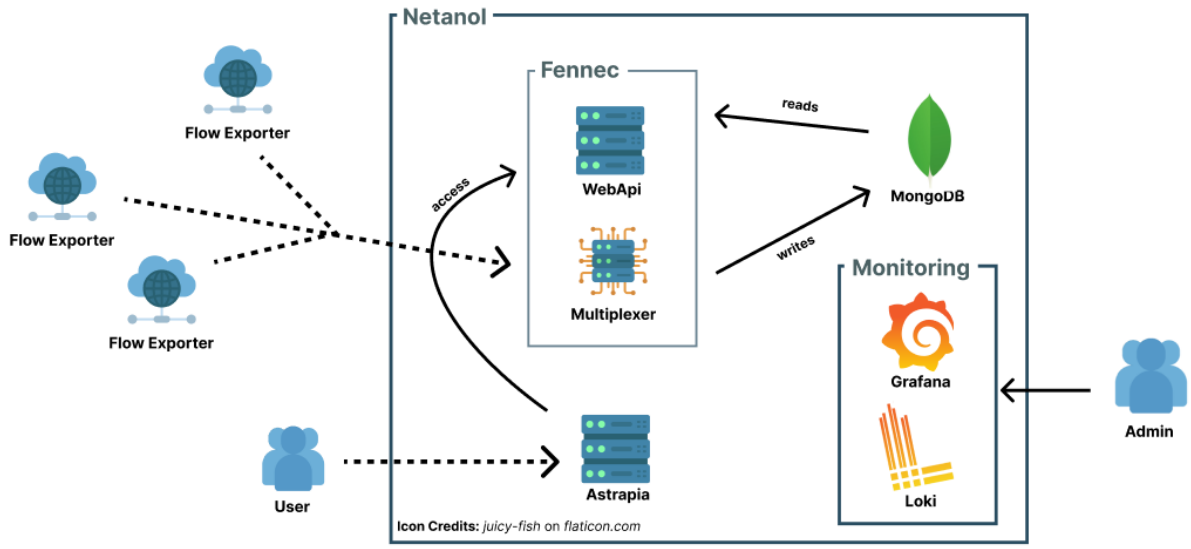
1.4 Product Scope

This analysis aids network administrators in comprehending network utilization, identifying issues, and optimizing performance. Netanol is designed with user-friendliness in mind, granting effortless access to information for well-informed decision-making. Customizable graphs and charts offer personalized perspectives on network utilization. Netanol enhances network performance, diminishes downtime, and boosts productivity by presenting a comprehensive overview of network traffic.

2. Overall Description

2.1 Product Perspective

Netanol is a new solution that serves as a specialized tool for network administrators, focusing on the collection, processing, and visualization of Flow data from network devices. This standalone product aims to increase transparency within networks, thereby offering opportunities for load optimization and troubleshooting through advanced analytics and graphing capabilities. Netanol aspires to be a comprehensive solution for understanding network performance, assisting administrators in making data-driven decisions for improved resource allocation. It does not replace existing systems; rather, it serves as a cost-effective alternative to current solutions.



2.2 Product Functions

Netanol will provide network visualization and analysis capabilities using various graphs. The major functions that the product must perform include:

- **Network visualization**: The product will be able to display the network structure using different graph layouts.

- **Data analysis**: The product will allow the user to analyse network data, such as identifying key nodes and edges, detecting patterns, and measuring network metrics.

Please refer to Section 4 (System Features) for more detailed information on these functions.

2.3 User Classes and Characteristics

Administrator

- Permission
  - Read, Write
- Security Level
  - High-level access without user management.
- Technical Expertise
  - Comprehensive understanding of network protocols, data analytics, and system administration.

User

- Permission
  - Read
- Security Level
  - Can only view data graphs but cannot make changes to the system.
- Technical Expertise
  - Basic level; just needs to understand how to navigate the dashboard.

2.4 Operating Environment

Netanol is designed with flexibility and compatibility in mind. Utilizing Docker for containerization, the software can run on virtually any system that meets Docker's requirements, making it OS-agnostic. For optimal performance, it is recommended to have a system with at least 4 cores, 8GB of RAM, and a 200GB SSD. This allows it to be deployed on a wide range of operating systems, including Windows, Linux, and macOS. Nevertheless, it is built to peacefully coexist with any existing network monitoring or management tools if such are already in place. This streamlined approach ensures hassle-free deployment and operation, irrespective of your existing software environment.

2.5 Design and Implementation Constraints

Certain limitations and constraints will influence the development of the Netanol project. In terms of hardware, Netanol's performance may be limited by the processing speed and memory capacity of the system where it is deployed. This becomes especially crucial when analysing large sets of Flow data

in real-time. Software-wise, the project is set to run in a Docker container, which could restrict it to environments where Docker can be installed and operated effectively. Technological constraints include the specific versions of frameworks and libraries that must be used. Finally, security considerations such as encryption standards and communication protocols must be adhered to, particularly since Netanol will deal with potentially sensitive network data.

2.6 User Documentation

For user documentation, as of the current stage, there is a Project Wiki that is maintained privately, hosting the essential technical details and usage guidelines for Netanol. The visibility and accessibility of this Wiki for a broader audience have not been finalized yet. Plans for comprehensive user manuals or integrated in-app help sections are still under discussion.

2.7 Assumptions and Dependencies

The entire application is reliant upon the export Flow traffic of network devices, rendering it an indispensable component. Without this crucial element, the application cannot perform any Flow analysis or displaying graphs.

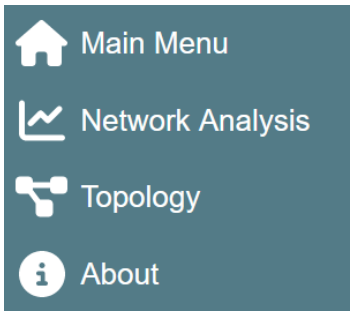
Domain	Technology	Version
Database	MongoDB	22.04 LTS (Jammy)
Deployment	Cloudflare	-
	Nginx	1.25.3
	Docker	24.0.6
API	C#	11
	ASP.NET	7.0.22
	.NET Core	7.0.22
	EF-Core	7.0.22
Frontend & Visualizing	NuxtJS	3.6.5
	NodeJs	18.17.1
	Axios	1.5.1
	Typescript	5.2.2
	D3	3.0.0
Monitoring	Grafana	10.2.3
	Loki	2.9.0

3. External Interface Requirements

3.1 User Interfaces

The admin dashboard of Netanol is crafted for straightforward navigation and ease of use. A sidebar serves as the main navigational guide, offering quick access to essential features like network monitoring and analytics. Central to the dashboard interface is the real-time network topology graph, which serves as an informational hub for network performance and

utilization. In addition to the graph, pie charts are available to offer detailed views on protocol usage, thus enhancing overall network transparency. This layout allows administrators to effortlessly manage and optimize their network.



Tab	Content Type
Main Menu	Landing Page
Network Analysis	Network statistic graphs
Topology	Topology visualization
About	Miscellaneous

3.2 Software Interfaces

Netanol is a high-performance NetFlow analyser designed to operate as a monolithic application. The system is exclusively focused on processing export data from network devices, eliminating any dependencies on third-party software components. At its core, the backend runs on C# (11), ASP.NET (7.0.22), and .NET Core (7.0.22), which work collaboratively to handle and analyse incoming NetFlow data. This data is then stored in a MongoDB (22.04 LTS) database for quick retrieval and further processing.

Upon receiving NetFlow exports, the data undergoes a series of analyses and transformations. The frontend, powered by NuxtJS (3.6.5) and NodeJS (18.17.1), retrieves this processed data through a REST API. Several libraries like Axios (1.5.1), TypeScript (5.2.2) and D3 (3.0.0) are employed to generate the visual representations of the analysed data.

Deployment is facilitated via Docker (24.0.6), making scalability and management straightforward. With this architecture, Netanol provides an end-to-end solution for NetFlow data visualization, offering both reliability and ease of use.

Incorporating Grafana (10.2.3) and Loki (2.9.0) into the Netanol project significantly enhances our monitoring capabilities. Grafana, with its dashboard features, will be utilized to visualize key metrics such as data processing speeds and system performance, offering real-time insights for educated decision-making. Loki's integration, as a log aggregation tool, complements this by efficiently managing and querying logs.

3.3 Communications Interfaces

The Netanol application interfaces only through web browsers and supports the popular browsers like Chrome (v95+) and Firefox (v89+). It is designed to operate using HTTP and HTTPS protocols for secure data transmission. On the network side, the system accepts NetFlow data, supporting multiple versions (v9, IPFIX, sFlow) of the NetFlow protocol to maximize compatibility with a range of network devices.

4. System Features

4.1 Traffic visualization between clusters.

4.1.1 Description and Priority

Provides a graphical representation of traffic flow between node groups (subnets, clusters, etc.). The width of the line between clusters will indicate the volume of traffic. This feature is of High priority.

4.1.2 Stimulus/Response Sequences

1. User opens the traffic visualization dashboard and is displayed with a graph containing nodes.
2. The topology tab displays traffic flow between node groups (subnets, clusters, etc.) with lines of varying widths based on traffic volume.

4.1.3 Functional Requirements

- REQ-1: Fetch real-time traffic data between node groups.
- REQ-2: Graphically represent traffic with lines between nodes.
- REQ-3: Adjust line width based on traffic volume.

4.2 Color Coding by Ports and Protocols

4.2.1 Description and Priority

Uses colour coding to differentiate traffic by ports and whether the protocol used is TCP or UDP. This feature is of medium priority.

4.2.2 Stimulus/Response Sequences

1. User views the traffic visualization topology dashboard.
2. The system displays traffic with color-coded lines based on port and protocol.

4.2.3 Functional Requirements

- REQ-1: Identify the port and protocol for each traffic Flow.
- REQ-2: Assign unique colour codes for different ports and protocols.
- REQ-3: Update the graphical interface to reflect these colour codes.

4.3 Analytics and Time Charts

4.3.1 Description and Priority

Displays analytics in the form of time charts for a specific time period as an optional feature. This is of medium priority.

4.3.2 Stimulus/Response Sequences

1. User selects a time range using a calendar.
2. The system displays traffic analytics as time charts for the selected time.

4.3.3 Functional Requirements

- REQ-1: Provide an option to select a time range.
- REQ-2: Generate time charts based on selected time range.
- REQ-3: Update the time charts in real-time for live data.

4.4 Firewall Connection and Rule Matching

4.4.1 Description and Priority

Determines which requests match with firewall rules and visualizes why a connection was allowed or blocked. This feature is of high priority.

4.4.2 Stimulus/Response Sequences

1. User views the traffic visualization dashboard.
2. The system highlights the connections that match firewall rules and provides a reason for allowing or blocking.

4.4.3 Functional Requirements

- REQ-1: Identify incoming and outgoing connections.
- REQ-2: Match connections against firewall rules.
- REQ-3: Visualize the reason for allowing or blocking a connection.

4.5 Node and Group Naming

4.5.1 Description and Priority

Allows users to give names to nodes and groups for easier identification. This is of low priority.

4.5.2 Stimulus/Response Sequences

1. User selects a node or a group.
2. User assigns a name to the selected node or group.

4.5.3 Functional Requirements

- REQ-1: Provide an option to select nodes or groups.
- REQ-2: Provide a text field to enter the name for the node or group.
- REQ-3: Save and display the custom names in the interface.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

Netanol operates in real-time to analyse Flows and visualize network traffic. As a network-critical application, it is expected to have a latency of less than 100 ms for displaying real-time analytics. The software should be able to handle large datasets efficiently and be capable of visualizing traffic for networks with up to ~2000 nodes. After stress-testing Netanol, it efficiently processed up to 200 packets per second, leading to the creation of about 600 new traces every second. These figures are based on real data, reflecting the system's real capabilities. Additionally, in these tests, the system demonstrated its quick data retrieval ability, reading between 100,000 to 600,000 traces in approximately one second.

5.2 Safety Requirements

Due to the nature of its operation within a network, Netanol must be designed to mitigate the risk of introducing vulnerabilities or creating performance bottlenecks in the network it is analysing.

5.3 Security Requirements

-

5.4 Software Quality Attributes

Well-structured and comprehensive documentation is essential for Netanol's ongoing development and maintenance. Good documentation serves as a roadmap, aiding new developers in understanding the system's architecture and logic, and facilitating easier debugging and feature addition. By making

this a priority, Netanol aims to ensure the software's long-term adaptability, maintainability, and ease of enhancement.

5.5 Business Rules

Only authorized network administrators should have access to Netanol’s full suite of features, including advanced analytics and configuration options. Regular users can view analytics but should not be able to alter configurations or view sensitive network information.