

# Segment Routing v6 (SRv6) Network Programming

Semester Work  
Autumn Term 2022

Date: 2022-12-23 10:48:17+01:00

**Authors:** Arnaud Kenzler  
Tsigereda Nebai Kidane

**Advisor:** Laurent Metzger

**Co-Advisor:** Severin Dellsperger



Computer Science Department

# Contents

<b>I</b>	<b>Abstract</b>	<b>1</b>
<b>II</b>	<b>Management Summary</b>	<b>3</b>
1	Management Summary	4
<b>III</b>	<b>Technical Documentation</b>	<b>7</b>
<b>2</b>	<b>Vision</b>	<b>8</b>
<b>3</b>	<b>Requirements</b>	<b>9</b>
3.1	Functional Requirements . . . . .	9
3.1.1	Use Cases . . . . .	10
3.2	(NFRs) - Non-Functional Requirements . . . . .	11
<b>4</b>	<b>Architecture</b>	<b>13</b>
4.1	Overview - C4 Model . . . . .	13
4.1.1	Frontend . . . . .	16
4.1.2	Backend-API . . . . .	17
4.1.3	Config-API . . . . .	19
4.1.4	PostgreSQL . . . . .	21
4.2	Design Decisions . . . . .	22
4.2.1	Docker Architecture . . . . .	22
4.2.2	ReactJs, Node, Postgres, Go . . . . .	22
4.2.3	YAML, Gonja2-Netconf, Scrapligo, Gornir . . . . .	23
4.2.4	UI Design . . . . .	23
4.3	Addressing Non-Functional Requirements . . . . .	23
4.4	Credentials . . . . .	24
4.5	External Interfaces . . . . .	25
4.5.1	Network . . . . .	25
4.5.2	Jalapeno Gateway API . . . . .	26
4.5.3	Jalapeno . . . . .	26

<b>5</b>	<b>Quality Measures</b>	<b>27</b>
5.1	Conventions . . . . .	27
5.1.1	Git Workflow . . . . .	27
5.2	CI/CD . . . . .	27
5.3	Quality Tracking . . . . .	28
5.3.1	Organizational . . . . .	28
5.3.2	Functional Requirements . . . . .	28
5.3.3	Non Functional Requirements . . . . .	29
5.4	Test Concept . . . . .	29
<b>IV</b>	<b>Project Documentation</b>	<b>30</b>
<b>6</b>	<b>Project Plan</b>	<b>31</b>
6.1	Processes, Meetings and Roles . . . . .	31
6.2	RUP Phases, Milestones, and Workitems . . . . .	32
6.2.1	RUP Phases and Milestones . . . . .	32
6.2.2	Workitems . . . . .	35
6.3	Planning Tools . . . . .	35
6.4	Risk Management . . . . .	36
<b>7</b>	<b>Time Tracking Report</b>	<b>37</b>
7.1	Reporting . . . . .	37
<b>8</b>	<b>Meeting Minutes</b>	<b>38</b>
<b>V</b>	<b>Appendix</b>	<b>39</b>
<b>9</b>	<b>Appendix</b>	<b>40</b>
9.1	Domain Model . . . . .	40
9.2	Moqups . . . . .	41
9.3	Time Tracking . . . . .	42
9.4	Administration . . . . .	61

**Part I**  
**Abstract**

# Segment Routing v6 network programming

## Students



Arnaud Kenzler



Tsigereda Nebai Kidane

**Initial Situation:** Configuring network devices using the command line interface (CLI) is usually time consuming because all commands have to be written for each function inside devices.

These commands must be applied to each device individually.

Furthermore, the specific commands have to be known and correct.

These reasons make CLI not very user friendly.

**Objective:** The goal of this work is to develop a web based SRv6 Network Programming Web App that allows a user to configure a given network. Configuring SR endpoint behavior, such as L3VPN on a network should be possible.

The user selects which configuration he/she wants to have on a specified router of the network, then the app configures the devices as desired.

The SRv6 Network Programming Web App only shows the ProviderEdge routers of the provided network topology, as only these routers are required to configure SRv6. Generally speaking, this project is designed to facilitate the configuration of SRv6 network programming on endpoints by providing a web app.

**Result:** A web app was developed with ReactJS, which allows L3VPN to be configured.

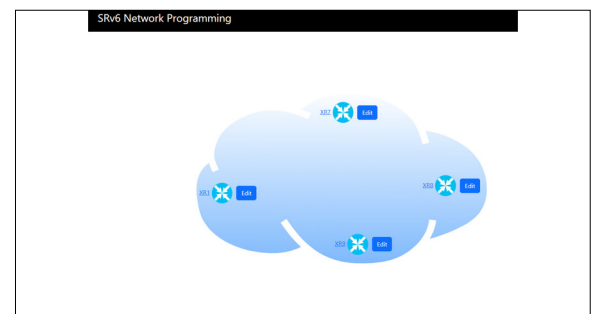
For this purpose, a single source of truth was built, which stores the data in a PostgresDB.

The connection to the frontend is done via an API based on GO.

To write the data from the single source of truth to the devices, a GO component was developed that uses Jinja-netconf, Gornir and Scraplgo.

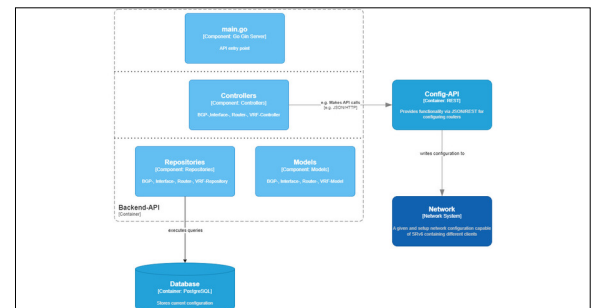
## Overview

Own presentation



## Backend

Own presentation



## Configuration template

Own presentation

The screenshot shows the SRv6 Network Programming web application interface. It includes fields for:

- Router:** XR1
- L3VPN:** A dropdown menu.
- VRF:** Cust-A
- Interface:** GigabitEthernet0/0/0
- VRF Name:** Cust-A
- RT Both:** 196
- Name:** to-Cust-A-LIS
- Description:** A
- IP Address:** 100.0.0.1
- Mask:** 255.255.255.252
- Current VRF:** Cust-A
- ASN:** 64097
- Router ID:** 1.1.1.1
- Neighbor IP:** 100.0.0.2
- Remote AS:** 64001

Buttons for 'Create New VRF', 'Create New Neighbor', 'Back', and 'Apply' are visible. A note at the bottom states: 'VRFs that are not used are automatically deleted. This also includes the BGP VRF neighbours.'

## Advisor

Prof. Laurent Metzger

## Subject Area

Internet Technologies and Applications

## Project Partner

Cisco Systems  
Belgium, Machelen,  
Belgium



**Part II**

**Management Summary**

# Chapter 1

## Management Summary

Configuring network devices using CLI could be time consuming because we have to write commands for each function we want to have in our device, that also means you need to have a good knowledge of the commands. The reasons mentioned here and others make CLI not user-friendly.

The SRv6 Network Programming app is a web based tool, which lets a user to be able to configure SR endpoint behaviors, such as L3VPN on a network. The user selects which configuration he/she wants to have on which router of the provided network and provide the necessary details, then the app will configure the device as requested. The SRv6 Network Programming app only displays the ProviderEdge (PE) routers of the network topology provided because only those routers are necessary for configuring SRv6. To generalize, this project was developed to make SRv6 network programming configuration on endpoints easier.

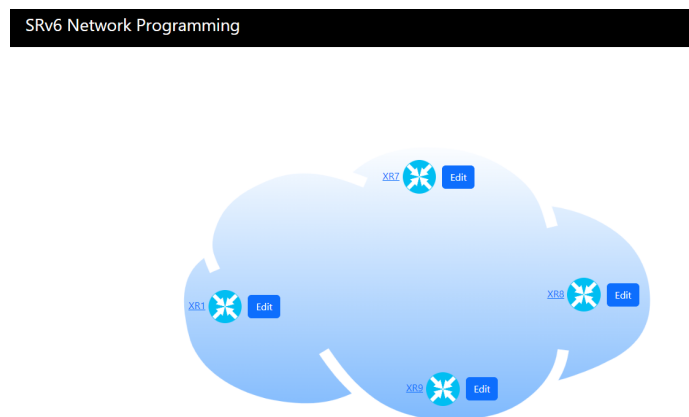


Figure 1.1: Main Page

To create a robust and easily extendable architecture, we decided to use a setup with docker containers. To keep the initial setup time to a minimum, deployment uses simple docker-compose capabilities. Creating configuration files to be deployed on larger scale Kubernetes clusters would be a logical improvement.

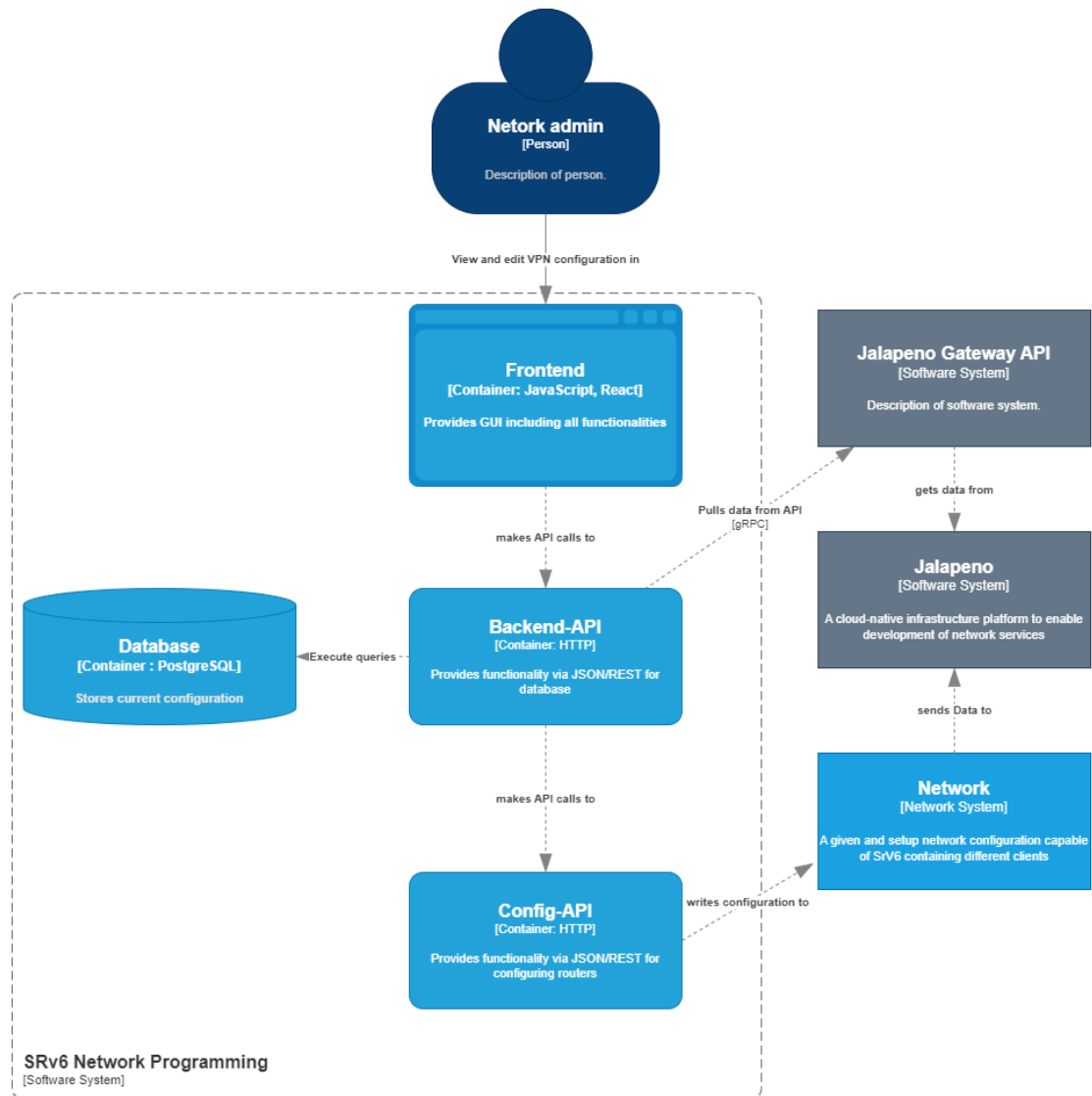


Figure 1.2: Container Architecture



The resulting product doesn't cover all the initially defined features because of shortage of time. At the moment the app can only configure L3VPN on the network provided, the other SR endpoint behaviors could be extended as a bachelor work.

## SRv6 Network Programming

XR1

L3VPN

**VRF**

Select VRF

**Interface**

Select Interface

VRF Name

RT Both

Description

Save VRF

---

**BGP**

ASN

64097

Router ID

1.1.1.1

**Neighbors**

Please select a VRF to display neighbors

Back

Apply

Figure 1.3: Edit Page

**Part III**

**Technical Documentation**

## Chapter 2

# Vision

Segment Routing v6 (SRv6) is the most promising technology for the transport of information in telecommunication networks. It is still in adoption phase and an application that can demonstrate all the typical services (like L3VPN, cross-connect, EVPN, etc.) that are currently standardized would greatly help the network community to familiarize with network programming with SRv6.

SRv6 uses a network programming approach that enables a network operator or an application to specify a packet processing program by encoding a sequence of instructions in the IPv6 packet header.

The goal of this project is to enable users to configure SR Endpoint Behaviors on their network devices, such as L3VPN, X-connect on a given network.

## Chapter 3

# Requirements

### 3.1 Functional Requirements

The use cases in the table 3.1 below show the functional requirements of the app. Detailed information and status of work is tracked centrally using Jira.

### 3.1.1 Use Cases

#	Goal	Brief	Criteria
1	Display current network	The app displays the configuration of the current network. Only the PE's and its attached networks are displayed.	Must
2	CRUD Configuration	The PE's of a given networks are displayed with its current state. The IP's of the networks are known and provided by the user. The state shows if the configuration has been written successfully to the devices. A new configuration can be created for a set of interfaces for a specific router inside the app and is stored inside a database before the configuration is written. The configuration of a PE can be deleted for a specific interface.	Must
2.1	Write configuration to the network	The app is able to write the configuration 2.3 to 2.5 from the app to the routers in the network. The network must already be able to handle SRv6 configurations.	Must
2.2	Read configuration from the network	The app is able to read the configuration 2.3 to 2.5 from the routers in the network. And update the state of the devices in the app.	Can
2.3	Configure DT4	The configuration can be set to DT4 between networks	Must
2.4	Configure DX2	The configuration can be set to DX2 between networks	Can
2.5	Configure DX2V	The configuration can be set to DX2V between networks	Can

Table 3.1: Use Cases

## 3.2 (NFRs) - Non-Functional Requirements

The prioritization of the following NFRs is according to their order. The smaller the NFR ID, the higher the priority. For MVP, we will focus on NFR-1, NFR-2 and NFR-3. We will check and discuss the NFRs in each sprint retrospective.

ID	NFR-1
Type	Deployment
Definition	The application needs to be deployable as a container
Mandatory	Yes

ID	NFR-2
Type	Availability
Definition	The application works regardless of whether central frontend is running or not.
Mandatory	Yes

ID	NFR-3
Type	Performance
Definition	The application gets data from the source (database or devices(if Jalapeno API would be used)) and displays them within an acceptable time. And it should also be able to configure devices within an acceptable time range.
Mandatory	Yes

ID	NFR-4
Type	Compatibility - Interoperability
Definition	The application is compatible with the latest version of central frontend.
Mandatory	Yes

ID	NFR-5
Type	Load Tolerability
Definition	Around 5 users can configure around 1000 PEs at the same time, and a user can not configure a device when there is already another configuration running.
Mandatory	Yes

ID	NFR-6
Type	Expendability
Definition	The application is able to read and write other SR endpoint behaviors from RFC 8986 such as: EVPN ... other than the ones used during development.
Mandatory	No

# Chapter 4

## Architecture

### 4.1 Overview - C4 Model

The following graphics 4.1 describes the system's architecture, which were planned at the early stage of development.

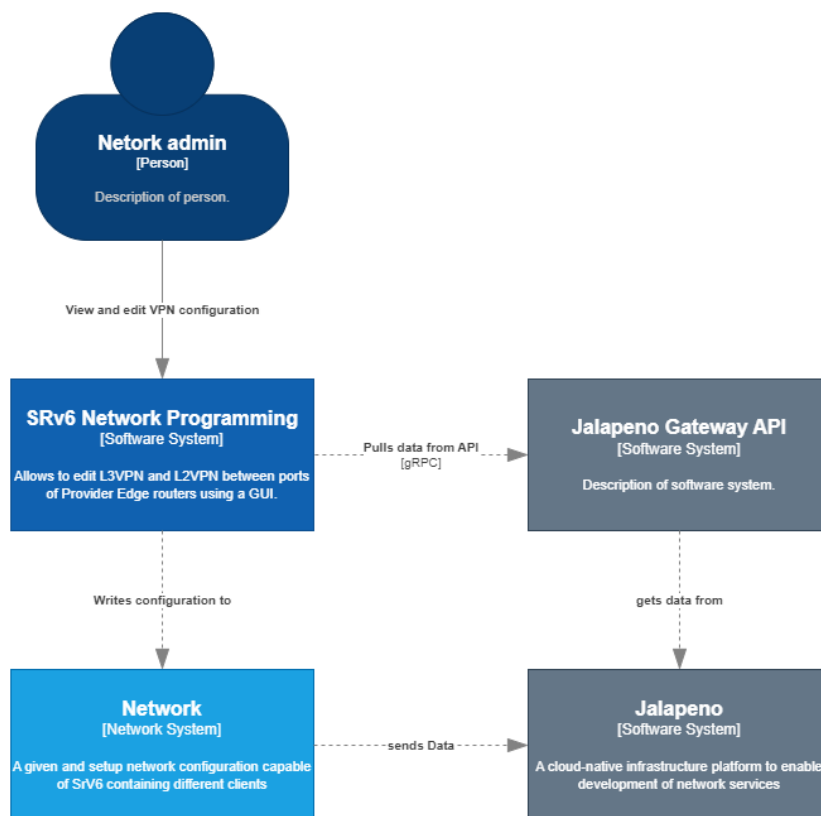


Figure 4.1: Context Architecture



Jalapeno and Jalapeno API are colored gray because they are not used in the application. The plan in the beginning was to use those components to get current state of the devices from the network, but we couldn't use those components because those components doesn't deliver the data we need in the application.

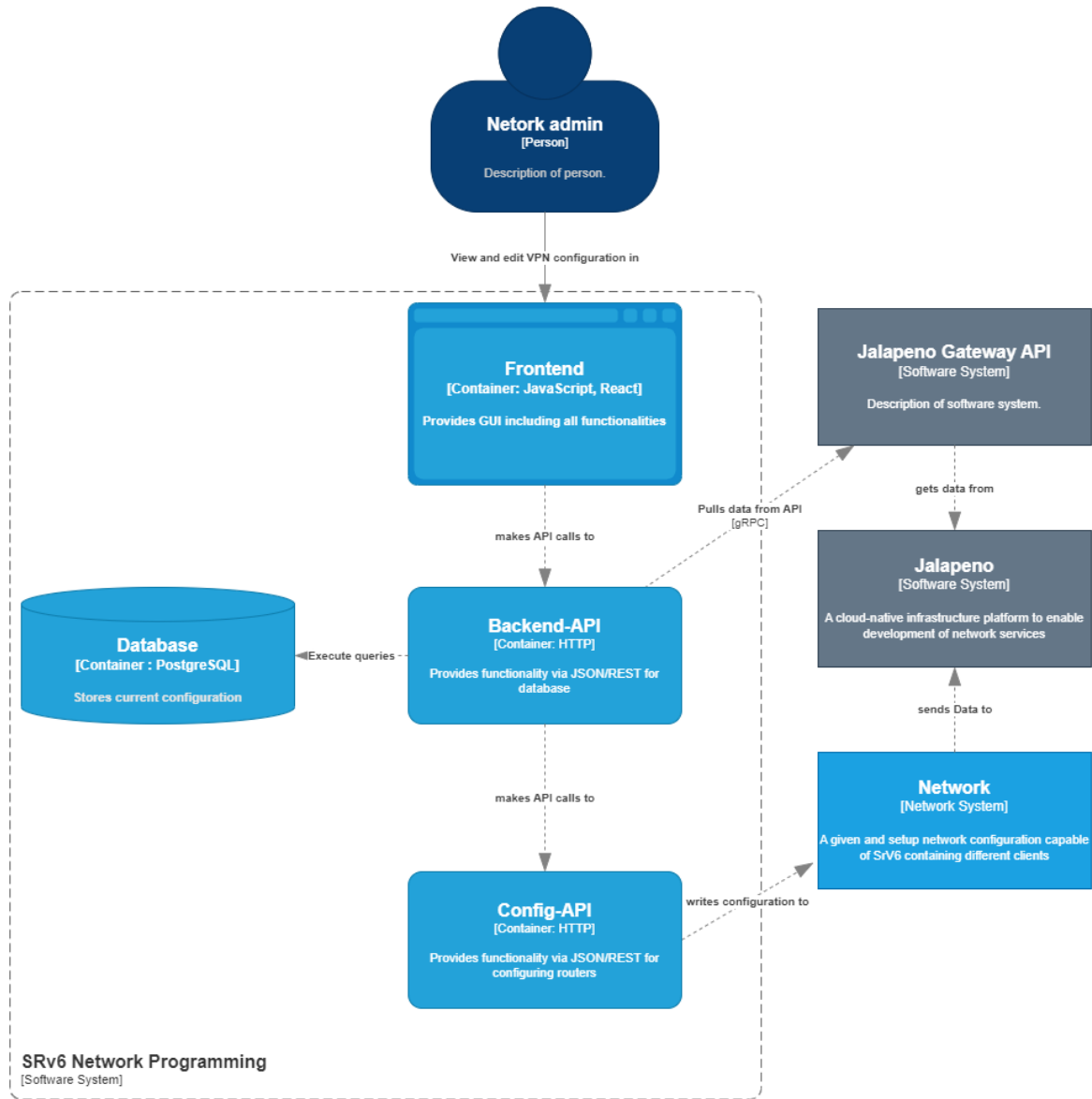


Figure 4.2: Container Architecture

Figure 4.2 shows how the components of the SRv6 Network Programming App are implemented. Named container architecture in C4 Model also shows the different containers we have in our application.

The SRv6-App is containerized using Dockerfiles and using a docker-compose file, we are able to have the same environment regardless of the machine we use. This leads to a flexible and easy setup.

Moreover, three different networks are used for each connection, this helps us to isolate the containers from each other.

- **Frontend-network:** Frontend to Backend-API
- **Backend-network:** Backend-API to Config-API
- **Database-network:** Backend-API to Database

<b>Container</b>	<b>Base Image</b>	<b>Extra</b>
Frontend	node:alpine	Custom image built with Dockerfile. Using ReactJs library
Backend-API	golang:alpine	Custom image built with Dockerfile
Config-API	golang:alpine	Custom image built with Dockerfile

### 4.1.1 Frontend

The Frontend component is the main React website, which lets users have an overview of the PE-routers of the network and enables them to configure L3VPN on the PE-routers. ReactJS was used to develop the frontend of the application. We used React with Javascript not Typescript because we both have experience with JavaScript.

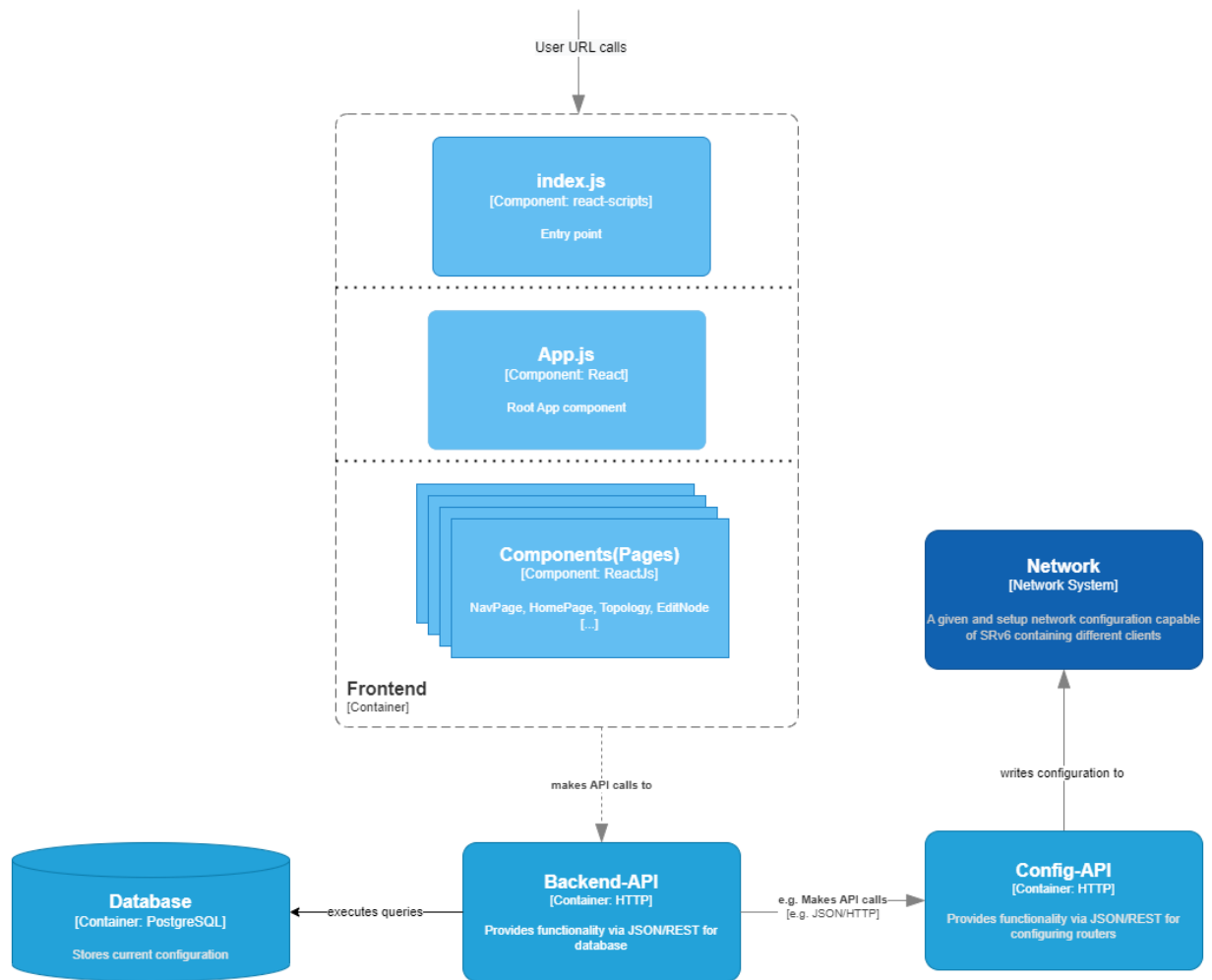


Figure 4.3: Frontend Component Architecture

Figure 4.3 shows the logic behind the Frontend container, such as its entry point and pages of the application.

### 4.1.2 Backend-API

The Backend-API is a Go(Gin) based REST API that allows CRUD operations on the PostgreSQL database.

The design chosen is similar to MVC, but in our case Model, Repository and Controller, which helps us to avoid direct access to the database and is also easier to containerize using docker. Moreover, within each directory we also create a separate file for each entity.

The Backend-API sits between almost all components and enables the communication between them. We separated the APIs into two: Backend- and Config-API to reduce complexity and separate functionality of the components. Additionally, the Backend-API prevents direct access to the database from the frontend- and/or config- component.

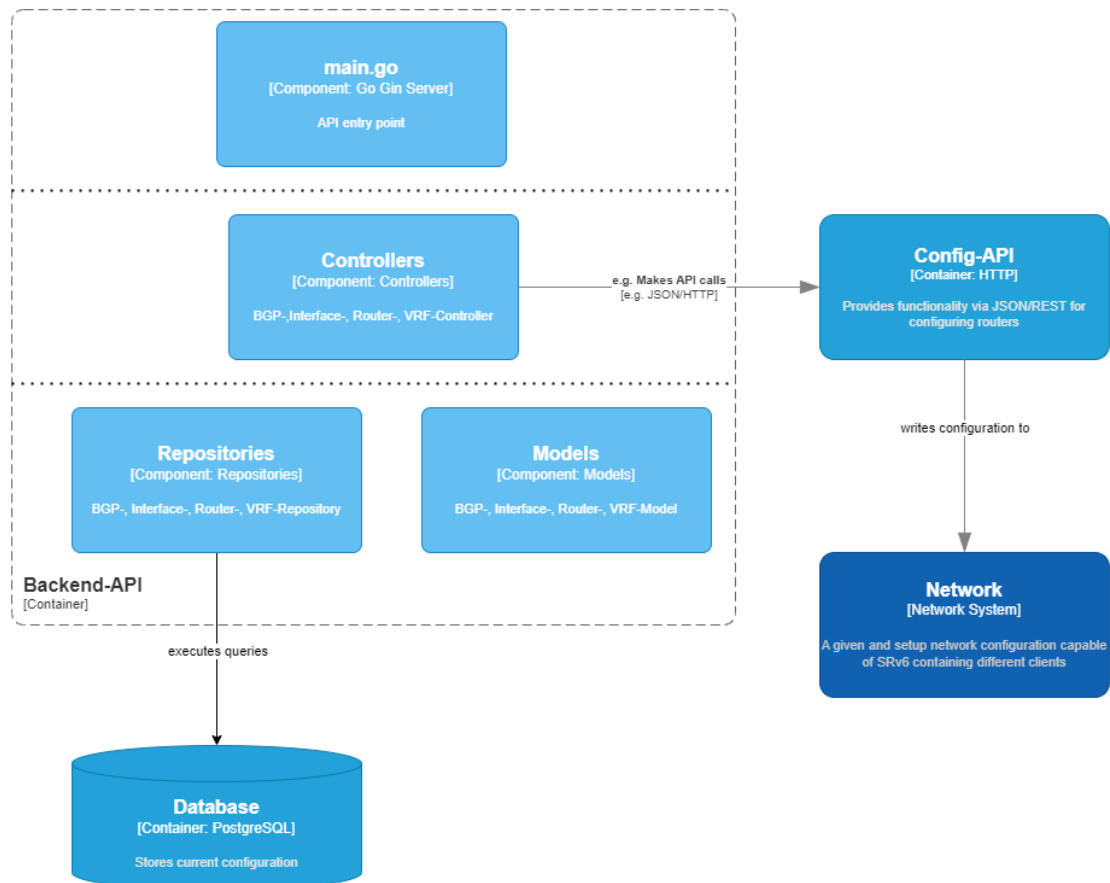


Figure 4.4: Backend-API Component Architecture

Figure 4.4 shows how the logic behind the Backend-API container looks like.

The following table 4.1 lists all API calls provided by the Backend-API component, that are currently being used in the application.

<b>Method</b>	<b>URL</b>	<b>Description</b>
GET	/api/routers	Gets all routers
GET	/api/routers/:routerid	Gets a router with the given id
GET	/api/bgps	Gets all the bgps in the database
GET	/api/routers/:routerid/bgp/	Gets the bgp of the router with the given id
GET	/api/vrfs	Gets all the vrfs in the database
GET	/api/vrfs/:vrfid	Gets a vrf with the given id
GET	/api/vrfs/:vrfid/name	Gets the name of a vrf with the given id
GET	/api/routers/:routerid/vrfs	Gets all the vrfs of a router with the given id
GET	/api/interfaces	Gets all the interfaces in the database
GET	/api/routers/:routerid/interfaces	Gets all the interfaces of a router with the given id
GET	/api/neighbors	Gets all the neighbors in the database
GET	/api/routers/:routerid/neighbors	Gets all the neighbors of a router with the given id
GET	/api/routers/:routerid/vrfs/:vrfname/neighbors	Gets all the neighbors of a router with the given id and which are under the vrf with the given vrfname
GET	/api/apply/:routerid	Applies the configuration to the router with the given id
POST	/api/vrf	Creates a new vrf
POST	/api/neighbors	Creates a new neighbor
POST	/api/routers/:routerid	Attaches the newly created vrf to the router with given id
PATCH	/api/vrfs/:vrfid	Updates the vrf with the given id with the changes made
PATCH	/api/routers/:routerid/interfaces/:interfacename	Updates the interface with the given interfacename of a router with the given id with the changes made
PATCH	/api/routers/:routerid/bgp/	Updates the bgp of a router with the given id with the changes made

Table 4.1: Backend-API Calls

The documentation for the Backend-API is automatically generated from annotations in the code. For this purpose swaggo is used. After deployment, the documentation is available at `/swagger/index.html`.

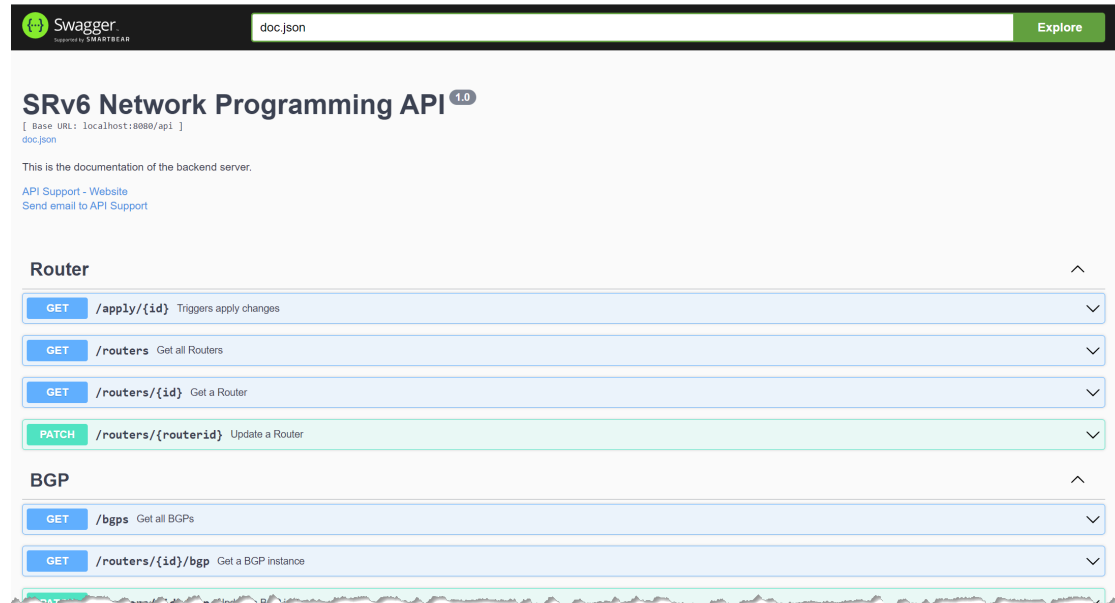


Figure 4.5: Swagger Documentation

### 4.1.3 Config-API

The Config-API is a Go(Gin) based REST API, which configures devices by getting the desired configuration from the database. We separated this in its own container to reduce complexity and dependency and be able to separate functionality.

The running-config of the PE router to be configured will be replaced by the edited version of a candidate-config.

The Config-API gets the running-config of the router and puts it in a candidate-config, this candidate-config will be modified using the given L3VPN configuration, and any unused vrfs will be deleted from the candidate-config. After that by committing the candidate-config, the running-config of the router will be replaced.

The Config-API sits between the Network and Backend-API components and enables the communication between those.

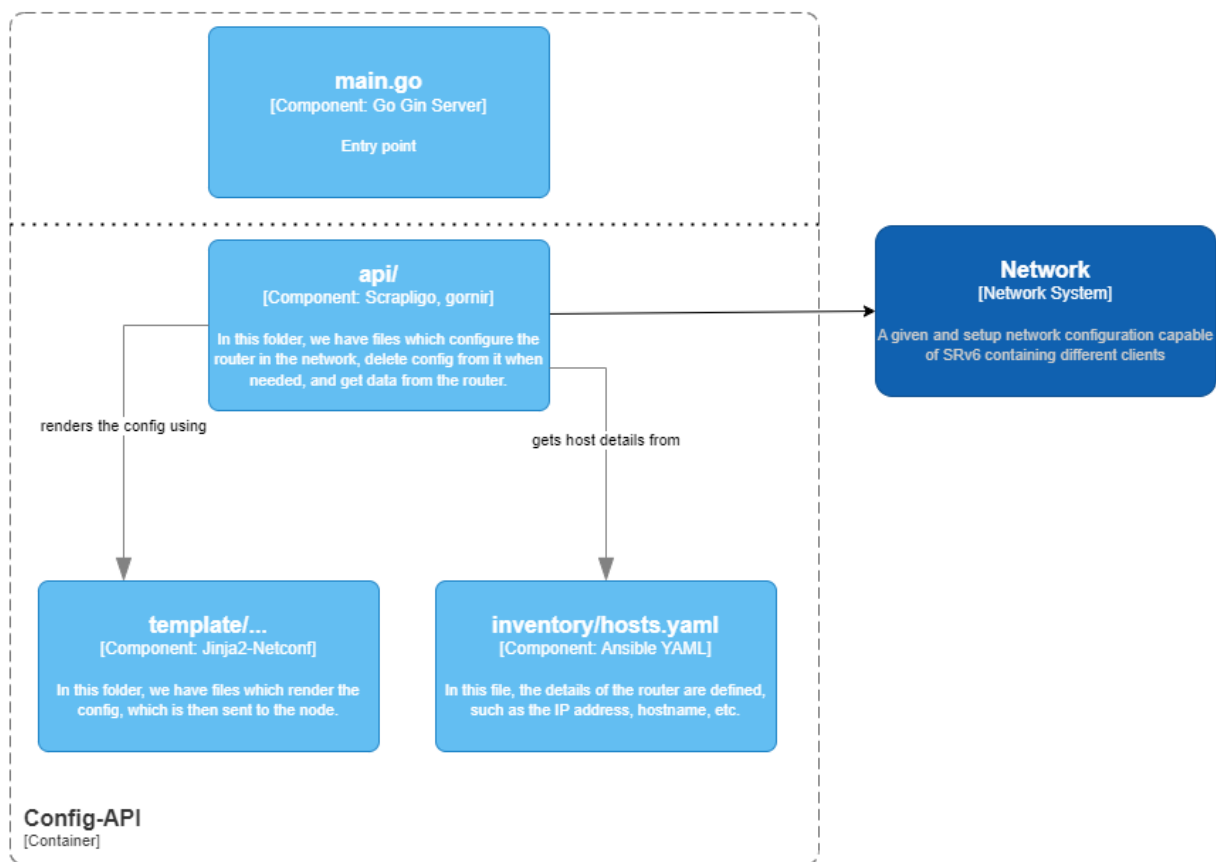


Figure 4.6: Config-API Component Architecture

Figure 4.6 shows how the logic behind the Config-API container, and how the file structure looks like.

The following table 4.2 lists all API calls provided by the Config-API component, that are currently being used in the application.

Method	URL	Description
GET	/node/update/:routerid	Applies the configuration to a router with the given id

Table 4.2: Config-API Calls

#### 4.1.4 PostgreSQL

The database is initialized using Gorm and the queries from Backend-API are also sent using Gorm. The Postgres database holds the desired state of the PE-routers. The idea of the project in the beginning was to use Jalapeno to get the data from the devices and use a database to save the desired state of the devices, but Jalapeno wasn't able to provide us the data (for example management IP of the devices, IPv4 of the links, etc.), which we need to configure the devices in the network. So we tried to solve this problem by initializing the data in the database manually. In an extended work of this project, this could be solved by using other tools like Netbox as a single point of truth.

Moreover, we are saving the configuration changes made on the PE router in the app in the database first, then apply the changes to the device by making an API call from the backend. The data to configure the router comes from the database not from the app directly. We do this because we wanted to have the database as a single point of truth and separate the APIs(Backend and Config) from each other to reduce complexity. This could be a problem because if the data saved in the database isn't correct, then the desired- and current-state of the router will be different, but the device won't be configured wrongly because the Config-API handles that in the backend.

There is also another option, we could try to apply the configuration to the router first, then save it in the database if it is correctly applied, but this could make our application slow, that's why we favored the first option. Those issues could be solved by using other single point of truth tools like Netbox as mentioned above.



## 4.2 Design Decisions

### 4.2.1 Docker Architecture

By implementing a dockerized setup, the goal is to increase flexibility and simplify deployment of the application. Docker-compose makes it relatively easy to set up without many dependencies or infrastructure needs.

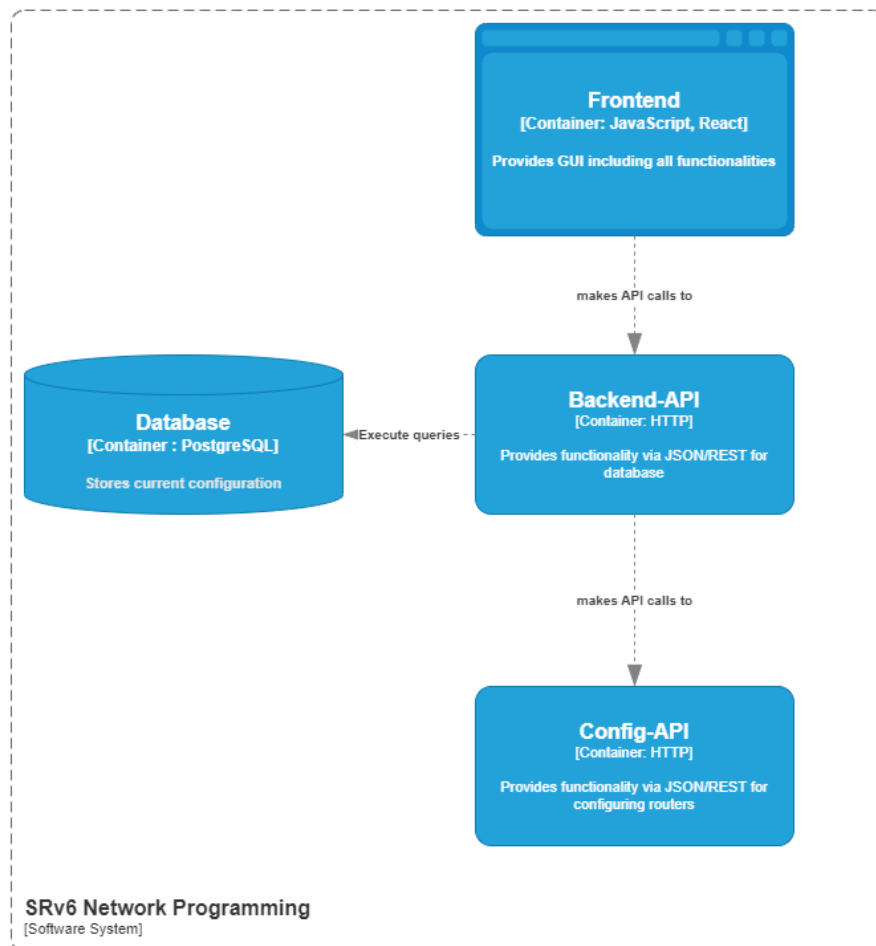


Figure 4.7: Docker Containers

### 4.2.2 ReactJs, Node, Postgres, Go

The product owner wanted that the app should be implemented using React (Frontend) and Go (Backend) to insure compatibility to central-frontend. We were open to choose other tools ourselves, so Postgres was used because the team members had prior experience with it.

Moreover, we used React with JavaScript not Typescript because we both had experience with JavaScript, so we wanted to save time by using the technology we had experience with.

### 4.2.3 YAML, Gonja2-Netconf, Scrapligo, Gornir

YAML and Gonja2-Netconf are used because the team members had prior experience with those tools. The product owners wish was to use Go whenever possible, so Scrapligo and Gornir, the Go version of nornir\_scrapli and nornir respectively, were favored because of their functionality and compatibility to the network devices in the provided demo network topology.

- **YAML:** Hosts(inventory) file is written in YAML. As mentioned above we used YAML because we had prior experience with it.
- **Gornir:** used to manage the hosts(inventory). The reason behind using gornir to manage the hosts instead of scrapligo is if the app was to be used to parallelly manage multiple devices in the future, then gornir is the best option.
- **Gonja-Netconf template:** Gonja is the Go version of Jinja. Used to render the configuration. As mentioned above we used gonja-netconf because we had prior experience with it.
- **Scrapligo:** used to connect to devices and configure them as wished by the user. We used scrapligo to configure the devices instead of gornir because gornir doesn't have the functions we wanted to use to send a template(gonja-netconf) rendered configuration to the devices.

We tried to use the tools we had experience with whenever possible because this helps us use our time efficiently to produce a functioning product.

### 4.2.4 UI Design

React-Bootstrap is used to design the application. We used ReactJS to implement the application, and we tried to use the libraries that come with it for example react-bootstrap, to avoid any incompatibility issues.

## 4.3 Addressing Non-Functional Requirements

- **NFR-1: Deployment**  
We use docker containers, which make it easy to deploy the application. The plan was to deploy the application in Kubernetes, but the time didn't let us do that, but in an extended work, the docker containers which are created now could be easily deployed in Kubernetes clusters.

- NFR-2: Availability  
At the moment the application can only be run locally, which means it isn't deployed anywhere. But we developed it without any dependability to any other applications such as central frontend, which means if it was going to be deployed, it would be available even if central frontend would crash.
- NFR-3: Performance  
To achieve this, we used Postgres as a single point of truth, which we don't wait long to get the data from the devices. Moreover, we also apply the configuration to the router by getting the desired state from the database, which will be done in the background, which increases the performance of our application, as it doesn't have to wait for the data to be written to the router and then update the database, and finally display the data from database.
- NFR-4: Compatibility - Interoperability  
We weren't able to integrate the application to central-frontend due to shortage of time, which means we weren't able to address this non-functional requirement. But integrating the application to central frontend could be done in extended work (BA), and this non-functional requirement could be addressed then.
- NFR-5: Load Tolerability  
The load tolerability of the application wasn't tested because we have no enough time to address it, but this would also be possible to be done in an extended work.
- NFR-6: Expendability  
As mentioned several times, the app is able to configure L3VPN on the devices, we weren't able to extend to other SR endpoint behaviors for the same reason as above: shortage of time.

## 4.4 Credentials

The credentials to access the database are stored inside a `.env` file inside the backend folder.

The credentials to login to the network devices is hard coded inside the `hosts.yaml` file inside `config/inventory` folder. We know this isn't a good idea, we could have used vault files for that, but we prioritized other issues. This could be implemented later if there is an extended work of this project.

## 4.5 External Interfaces

### 4.5.1 Network

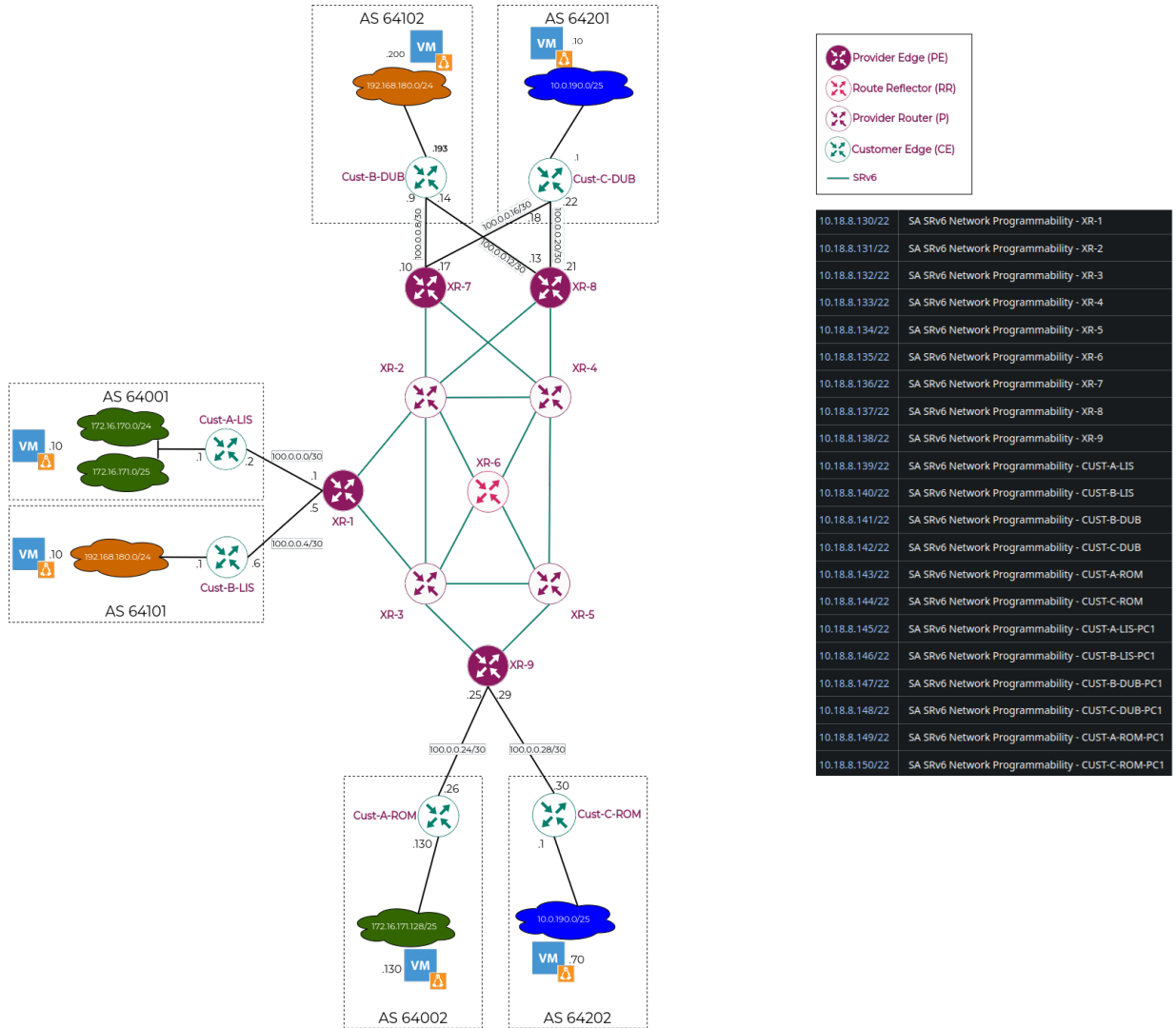


Figure 4.8: Network

A demo network topology, which was provided for the purpose of the project by the product owner. The driver of the routers in this demo network is Cisco IOS-XR, and in this project we only considered the PE (ProviderEdge) routers because the app does configure SR Behavior and for that purpose we only need the PE routers. The demo-network was already configured with the configurations, we only needed to configure the SR endpoint behaviors (L3VPN) on its PE nodes.

## 4.5.2 Jalapeno Gateway API

The idea of the project was to use the Jalapeno Gateway API to pull data from the network, but this wasn't possible because Jalapeno Gateway API isn't capable of getting the data (for example interfaces with their details), which were needed for our project, so we had to initialize the data manually into the Postgres database.

## 4.5.3 Jalapeno

The network devices are configured to send data to Jalapeno and as mentioned above in 4.5.2 we wanted to get those data using Jalapeno Gateway API, but for the same reason mentioned above in 4.5.2, we weren't able to use it.

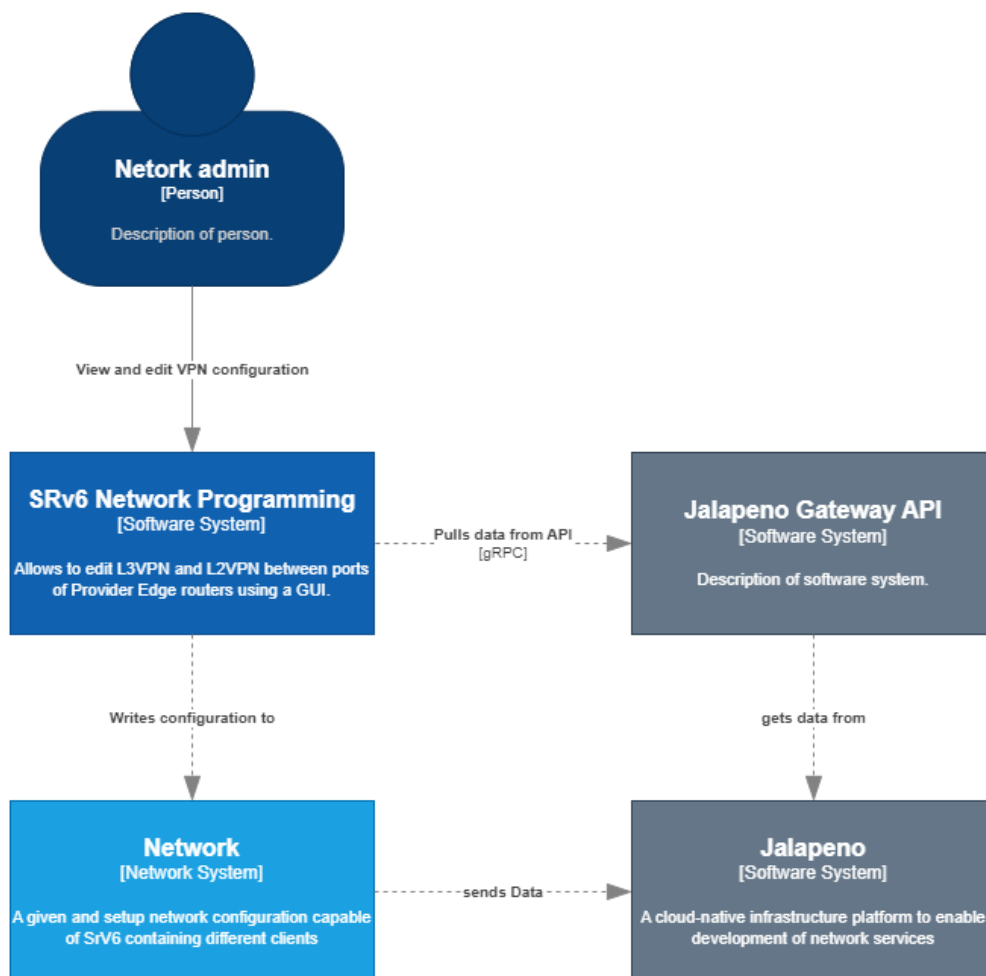


Figure 4.9: Context Architecture

# Chapter 5

## Quality Measures

### 5.1 Conventions

#### 5.1.1 Git Workflow

GitLab is used as a central repository (GitLab Repository), the following strategy is used:

- Branch *main* shall be used for merging finished functionality and small, self-contained changes.
- Branch *wip-documentation* shall be used for documentation updates.
- User Stories are developed on separate branches and merged into main sporadically when some functionalities are completed.

### 5.2 CI/CD

The current CI/CD Pipeline in GitLab can be used to execute the build tasks. It contains the following stages:

Stage	Jobs	Description
build	build_doc	Build Job for documentation using the provided makefile, store result as GitLab artifact.
diff	build_doc_diff	Build Job for documentation for manually triggering the action, store result as GitLab artifact.

The deployment of the application has not been automated due to shortage of time.

## 5.3 Quality Tracking

### 5.3.1 Organizational

Topics discussed in meetings and reviews are recorded in the meeting minutes (see chapter 8). Weekly meetings are held to discuss progress and problems that have arisen.

### 5.3.2 Functional Requirements

The quality of the functional Requirements will be tracked on the Jira Roadmap.

The following guidelines describe the process of implementing new requirements:

- Every user story is defined in smaller tasks, description should be added to the task if the task title isn't clear enough.
- Each team-member is responsible for reviewing the criteria of their assigned tasks. Further refinement or clarifying should be done if necessary.

#### Definition of Ready

To start a story or task, the following criteria must be met:

- A user story is created under its milestone. May split into tasks if the story is too big to be done at once.
- Unclear issues are discussed during meetings.

#### Definition of Done

To declare an issue as done, the following criteria must be met:

- All the necessary design and requirements information should be available, and the behavior should be tested.
- No known bugs that affect the app's functionality.

The above criterias apply to all issues in the user cases.

### **5.3.3 Non Functional Requirements**

In order to measure the quality of the NFR, different methods are used

- NFR-1, Deployment: Docker best practices has been applied to fulfill this requirement.
- NFR-2, Availability: Due to shortage of time, the app is anyway developed as standalone app, which means it is not yet integrated to central-frontend.
- NFR-3, Performance: This will be tested periodically during development.
- NFR-4 to NFR-6: will not be pursued due to prioritization and shortage of time.

### **5.4 Test Concept**

Due to lack of time we were not able to write very extensive tests. The basic functionality of the backend API is covered by tests in Postman. During the development we did some manual tests development usability and API tests.



**Part IV**

**Project Documentation**

# Chapter 6

## Project Plan

### 6.1 Processes, Meetings and Roles

#### Processes

As the time we have for the project isn't much, we think it is ideal to use Scrum+ (RUP (Rational Unified Processes) and Scrum combined), RUP as a planning tool for the whole project and Scrum inside each RUP iterations.

#### Meetings

Our meetings will be held on Tuesday and Friday afternoons. If there are issues that occur between meetings, Microsoft Teams will be the tool we will be using. After each milestone, we'll have a sprint retrospective, which will be taking about an hour. In this meeting, we will discuss the inputs from the review meetings. We will also track our risks. Finally, we will plan the next sprint, discuss our issues.

#### Sprints

Sprints will be taking two weeks.

**Daily** Daily stand-ups won't be used since both of us will be working on Tuesday and Friday together.

**Weekly Meeting** Team members and advisors will meet weekly on Tuesday 17:00-18:00.

**Sprint Retrospective** Team members discuss after each review meeting on Friday 13:00-14:00 on the sprint and the feedbacks of the review meeting.

**Sprint Review** This meeting will be held on Tuesday 14:00-15:00. The team members will talk about what they have done in the last sprint, and discuss the problems they had.

**Sprint Planning** The next sprint will be planned and workloads will be estimated. This takes place after sprint review.

## Roles

Because of the size of our project, we have decided that we don't need the role scrum master. Instead, we distribute the tasks among the team members. Only a project manager and an actuary are designated. Some tasks are not yet defined. The following roles have been assigned to team members, other than everyone being a developer:

- Project Owners: Laurent Metzger & Severin Dellsperger
- Project Manager: Arnaud Kenzler  
The Project Manager is responsible for leading through meetings, time and issue tracking
- Documentation/LaTeX: Tsigereda Nebai Kidane  
Responsible for maintaining Product and Project Documentation in LaTeX.
- Architecture: Tsigereda Nebai Kidane & Arnaud Kenzler
- Developers: Tsigereda Nebai Kidane & Arnaud Kenzler

## 6.2 RUP Phases, Milestones, and Workitems

Following are the phases and milestones we set up for this project.

### Dates and Figures

Start: 20.09.2022

Working days: Together (Tuesday(15:00-17:00) and Friday(13:00-17:00))

Resources: 480h (2Members \* 8ECTS \* 30h)

End: 23.12.2022

### 6.2.1 RUP Phases and Milestones

#### Inception - (1 Week)

**Milestone 1:** Planning - (27. September)

All relevant meetings will be scheduled on Tuesday and Friday afternoons. For communications or feedbacks between meetings, Microsoft Teams will be used.

- Project plan documented:  
containing milestones, project planning tools and meetings planned
- Project tools setup:  
time tracking, issue tracking, etc.
- Risks discussed and documented
- Project created in GitLab
- Rough FRs and NFRs discussed and documented

### **Elaboration - (3 Weeks)**

#### **Milestone 2:** Requirements - (*until 04. October*)

- Risks discussed
- Wireframe created
- Our machines fully setup (all the necessary tools installed)
- CI/CD initial config
- Functional and non-functional requirements finalized

#### **Milestone 3:** Architecture and Infrastructure - (*until 25. October*)

- Architecture created and documented
- User stories defined for the next phase
- Running prototype built

### **Construction - (8 Weeks)**

#### **Milestone 4:** MVP Release - (*until 08. November*)

- Working application according to MVP
- Tests are documented

#### **Milestone 5:** First Release(Beta Release) - (*until 29. November*)

- Most important functional and non-functional requirements are implemented
- Working application

#### **Milestone 6:** Final Release - (*until 13. December*)

- All features implemented and ready for release
- Bugs fixed
- Application tested

**Transition - (1 Week)**

**Milestone 7:** Final Documentation - (*until 20. December*)

- Application is working as wished by the product Owner
- Everything is well documented
- Abstract is written (Deadline: 19.12.2022)
- Prepare and hold presentation

**Milestone 8:** Submission - (*until 23. December*)

- Proof read the documentation
- Upload all files

**Timeline:** The figure below 6.1 shows the timeline of the SRv6 Network Programming project ...

## SRv6 Network Programming Timeline

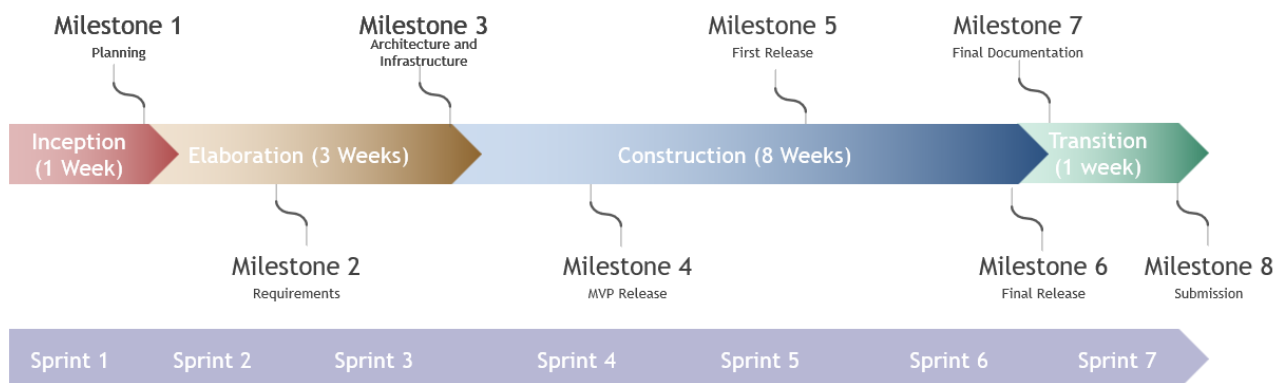


Figure 6.1: Timeline

**Roadmap:** The figure below 6.2 shows the roadmap of the SRv6 Network Programming project in Jira looks like this...

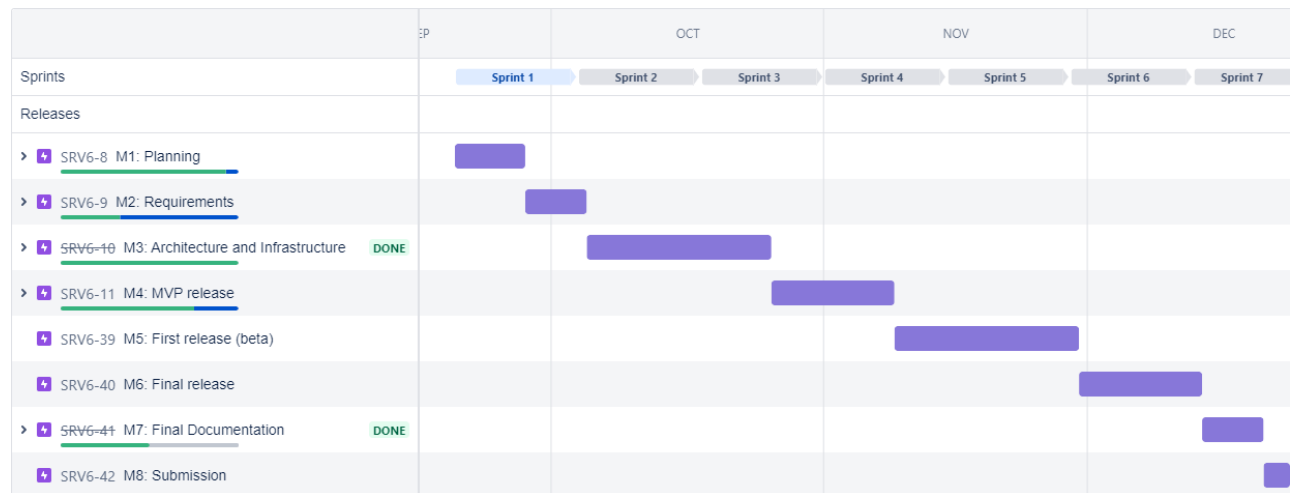


Figure 6.2: Roadmap

## 6.2.2 Workitems

Detailed work items and their current status are tracked in Jira. <https://srv6-sa.atlassian.net/>

**We have the following setup(order):**

Milestones → Epic → Stories → Task

## 6.3 Planning Tools

- Issue tracker: Jira will be used as issue tracking tool for any issues and standard tasks to be done within the project.
- Time tracker: Clockify is free and offers simple but efficient task tracking, tagging and reporting.
- Version Control: GitLab will be used.

## 6.4 Risk Management

The risks listed below might occur in the project.

Title	Description	Damage in hours	Probability	Weighted damage in hours	Prevention	Mitigation until
Technology risks	Lack of experience could be a big risk in our project.	80	50%	40	We will have to plan time to learn new technologies.	End of Elaboration
Tight schedules	As we are taking other modules beside this semester work, it could sometimes get a bit tight.	20	10%	2	Focus enough time on planning ahead and making time to work on the project tasks. In the planned meetings we will discuss the next important tasks and prioritize the most important tasks.	End of Elaboration
Absence of Team Members	Team member could get sick or some emergency absence happens.	20	10%	2	For short time absence of the team member will be informed about his tasks. In case of a long term absence(sickness), should be discussed with the advisors	Undefined
Conflicts between Team Members	Conflicts can happen between team members	10	10%	1	Focus on frequent communication. Try talking openly about problems that may arise.	Undefined
GitLab Outage	GitLab doesn't work	20	10%	2	Frequent pushing and pulling of everyone's local git repository minimizes the risk of losing data, try to talk with the one responsible as soon as possible.	Undefined
Automation Tools Incompatibility	Automation tools with go are mostly new and there could be compatibility problem.	50	20%	10	Ask the for help as soon as possible	Undefined

# Chapter 7

## Time Tracking Report

### 7.1 Reporting

We use clockify as our time tracking tool. You can find the reports here: [Clockify](#).

You will also find detailed info in this directory ("[/appendix/clockifyTimeTracking](#)")

Total summary of the time divided by Phase and project member:

<b>Phase</b>	<b>Arnaud Kenzler</b>	<b>Tsigereda Nebai Kidane</b>	<b>Total used</b>	<b>Time remaining</b>
Inception	13:46	19:53	33:39	446:21
Elaboration	33:13	59:12	92:25	353:56
Construction	84:34	131:29	216:03	137:53
Transition	34:02	23:24	57:26	80:27
<b>Total</b>	<b>165:35</b>	<b>233:58</b>	<b>399:33</b>	<b>80.27</b>



## Chapter 8

# Meeting Minutes

Meeting minutes are kept in Microsoft OneNote: Project Team Notebook SharePoint

Part V  
Appendix

# Chapter 9

## Appendix

### 9.1 Domain Model

The graphic 9.1 below shows the database model of the application.

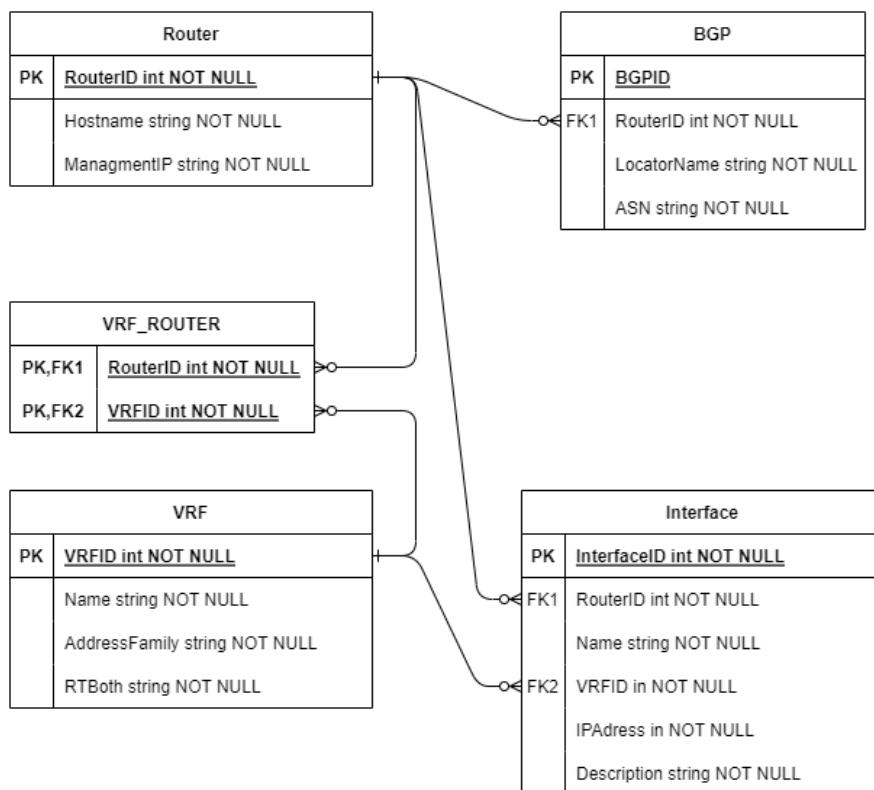


Figure 9.1: Database Model

## 9.2 Moqups

Figure 9.2 and 9.3 show the Moqups design of the app.

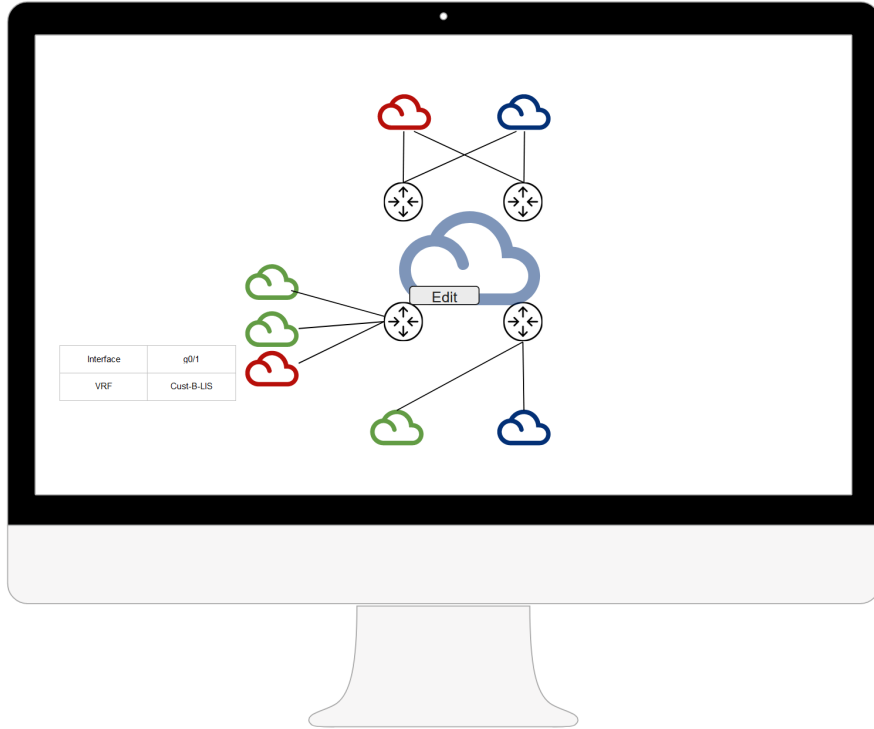


Figure 9.2: Home Page Moqups

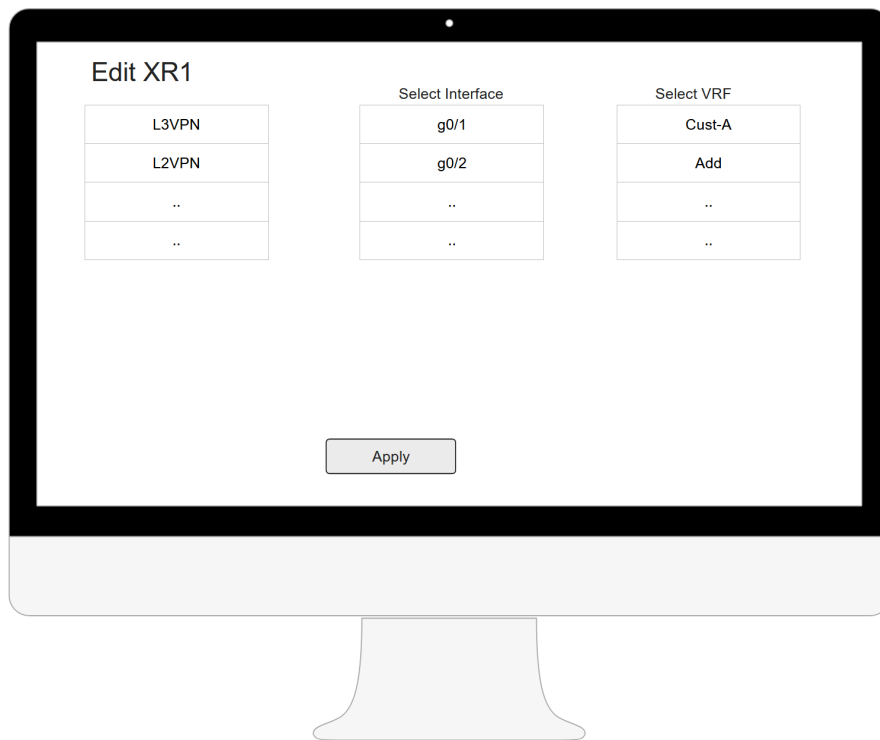


Figure 9.3: Edit Page Moqups

### 9.3 Time Tracking

The files below show the detailed overview of the time tracking from Clockify.

# Detailed report

16/09/2022 - 27/09/2022



Total: 33:39:15 Billable: 33:39:15 Amount: 0.00 USD

Date	Description	Duration	User
27/09/2022	(Without Description) SRv-6 - [Meeting]	01:43:56 16:04:46 - 17:48:42	Arnaud Kenzler 0.00 USD
27/09/2022	Meeting SRv-6 - [Documentation, Project Planning]	02:48:16 15:00:00 - 17:48:16	Tsigereda Nebai 0.00 USD
27/09/2022	SRV6-14 SRv-6 - [Requirements]	01:14:36 14:50:03 - 16:04:39	Arnaud Kenzler 0.00 USD
27/09/2022	Reading SRv-6 - [Requirements]	00:46:31 11:44:08 - 12:30:39	Arnaud Kenzler 0.00 USD
27/09/2022	SRV6-14 SRv-6 - [Requirements]	00:23:24 11:20:31 - 11:43:55	Arnaud Kenzler 0.00 USD
27/09/2022	SRV6-13 SRv-6 - [Project Planning]	01:11:04 10:08:57 - 11:20:01	Arnaud Kenzler 0.00 USD
26/09/2022	Update README file SRv-6 - [Documentation]	00:16:52 15:40:00 - 15:56:52	Tsigereda Nebai 0.00 USD
26/09/2022	Issue #SRV6-15 - NFRs SRv-6 - [Documentation, Requirements]	00:17:38 15:03:25 - 15:21:03	Tsigereda Nebai 0.00 USD
26/09/2022	Issue #SRV6-15 - NFRs SRv-6 - [Documentation, Requirements]	00:53:08 13:46:52 - 14:40:00	Tsigereda Nebai 0.00 USD
26/09/2022	Issue #SRV6-15 - NFRs SRv-6 - [Documentation, Requirements]	00:13:38 12:10:00 - 12:23:38	Tsigereda Nebai 0.00 USD
26/09/2022	Issue #SRV6-15 - NFRs SRv-6 - [Documentation, Requirements]	00:16:09 11:43:40 - 11:59:49	Tsigereda Nebai 0.00 USD
25/09/2022	Issue #SRV6-15 - NFRs SRv-6 - [Requirements]	00:25:00 15:00:00 - 15:25:00	Tsigereda Nebai 0.00 USD
23/09/2022	(Without Description) SRv-6 - [Meeting]	00:50:28 15:41:58 - 16:32:26	Arnaud Kenzler 0.00 USD
23/09/2022	(Without Description) SRv-6 - [Project Planning]	00:30:27 15:11:26 - 15:41:53	Tsigereda Nebai 0.00 USD

23/09/2022	(Without Description) SRv-6 - [Meeting]	01:10:25 13:34:06 - 14:44:31	Arnaud Kenzler 0.00 USD
23/09/2022	Meeting SRv-6 - [Meeting]	01:25:34 13:15:00 - 14:40:34	Tsigereda Nebai 0.00 USD
23/09/2022	Issue #SRV6-38 and Issue #SRV6-37 SRv-6 - [Documentation]	01:02:04 10:59:58 - 12:02:02	Tsigereda Nebai 0.00 USD
23/09/2022	Issue #SRV6-38 and Issue #SRV6-37 SRv-6 - [Documentation]	00:39:32 10:15:00 - 10:54:32	Tsigereda Nebai 0.00 USD
23/09/2022	Issue #SRV6-36 SRv-6 - [Project Planning]	01:00:00 09:15:00 - 10:15:00	Tsigereda Nebai 0.00 USD
22/09/2022	Issue #SRV6-3 SRv-6 - [Project Planning]	00:16:47 16:30:00 - 16:46:47	Tsigereda Nebai 0.00 USD
22/09/2022	Issue #SRV6-23 SRv-6 - [Project Planning]	00:09:39 16:22:18 - 16:31:57	Tsigereda Nebai 0.00 USD
22/09/2022	Issue #SRV6-23 SRv-6 - [Project Planning]	00:11:13 15:48:11 - 15:59:24	Tsigereda Nebai 0.00 USD
22/09/2022	Issue #SRV6-20 SRv-6 - [Project Planning]	00:47:38 15:00:00 - 15:47:38	Tsigereda Nebai 0.00 USD
22/09/2022	Issue #SRV6-20 SRv-6 - [Project Planning]	00:42:06 10:58:45 - 11:40:51	Tsigereda Nebai 0.00 USD
21/09/2022	Issue #SRV6-5 SRv-6 - [Documentation, Project Planning]	03:03:53 21:45:00 - 00:48:53	Tsigereda Nebai 0.00 USD
21/09/2022	(Without Description) SRv-6 - [Requirements]	00:54:41 17:38:40 - 18:33:21	Arnaud Kenzler 0.00 USD
21/09/2022	Issue #SRV6-5 SRv-6 - [Documentation, Project Planning]	01:45:54 15:00:00 - 16:45:54	Tsigereda Nebai 0.00 USD
21/09/2022	(Without Description) SRv-6 - [Project Planning]	01:45:42 10:09:11 - 11:54:53	Arnaud Kenzler 0.00 USD
20/09/2022	Meeting SRv-6 - [Meeting]	00:50:00 17:00:00 - 17:50:00	Tsigereda Nebai 0.00 USD
20/09/2022	(Without Description) SRv-6 - [Meeting]	00:45:00 17:00:00 - 17:45:00	Arnaud Kenzler 0.00 USD
20/09/2022	Project planning SRv-6 - [Documentation, Project Planning]	02:18:00 14:37:00 - 16:55:00	Tsigereda Nebai 0.00 USD

---

20/09/2022

(Without Description)  
SRv-6 - [Project Planning]

03:00:00  
14:00:00 - 17:00:00

Arnaud Kenzler  
0.00 USD



# Detailed report

28/09/2022 - 25/10/2022



Total: **92:24:37** Billable: **92:24:37** Amount: **0.00 USD**

Date	Description	Duration	User
25/10/2022	Meeting SRv-6 - [Meeting, Prototype]	01:07:29 16:42:05 - 17:49:34	Tsigereda Nebai 0.00 USD
25/10/2022	Meeting SRv-6 - [Meeting]	01:05:00 16:40:00 - 17:45:00	Arnaud Kenzler 0.00 USD
25/10/2022	Prototype SRv-6 - [Prototype]	03:00:32 13:07:24 - 16:07:56	Tsigereda Nebai 0.00 USD
25/10/2022	api SRv-6 - [Prototype]	07:30:00 09:10:00 - 16:40:00	Arnaud Kenzler 0.00 USD
24/10/2022	Prototype SRv-6 - [Prototype]	01:03:48 20:39:15 - 21:43:03	Tsigereda Nebai 0.00 USD
24/10/2022	Prototype SRv-6 - [Prototype]	01:40:37 15:49:23 - 17:30:00	Tsigereda Nebai 0.00 USD
24/10/2022	Prototype SRv-6 - [Prototype]	01:09:05 14:33:29 - 15:42:34	Tsigereda Nebai 0.00 USD
24/10/2022	Prototype SRv-6 - [Prototype]	00:32:10 13:04:00 - 13:36:10	Tsigereda Nebai 0.00 USD
24/10/2022	Prototype SRv-6 - [Prototype]	00:50:16 11:12:39 - 12:02:55	Tsigereda Nebai 0.00 USD
23/10/2022	SRV6-66 SRv-6 - [Prototype]	04:55:52 12:43:06 - 17:38:58	Arnaud Kenzler 0.00 USD
23/10/2022	Prototype SRv-6 - [Prototype]	00:17:06 11:50:00 - 12:07:06	Tsigereda Nebai 0.00 USD
22/10/2022	Prototype SRv-6 - [Prototype]	01:03:27 16:24:03 - 17:27:30	Tsigereda Nebai 0.00 USD
21/10/2022	Prototype SRv-6 - [Prototype]	00:49:19 14:55:21 - 15:44:40	Tsigereda Nebai 0.00 USD
21/10/2022	Prototype SRv-6 - [Prototype]	02:00:19 12:48:22 - 14:48:41	Tsigereda Nebai 0.00 USD

21/10/2022	Prototype SRv-6 - [Prototype]	00:07:48 12:27:16 - 12:35:04	Tsigereda Nebai 0.00 USD
21/10/2022	Prototype SRv-6 - [Prototype]	01:43:28 10:33:48 - 12:17:16	Tsigereda Nebai 0.00 USD
21/10/2022	Prototype SRv-6 - [Prototype]	01:15:15 09:15:00 - 10:30:15	Tsigereda Nebai 0.00 USD
20/10/2022	Prototype SRv-6 - [Prototype]	00:36:12 22:09:51 - 22:46:03	Tsigereda Nebai 0.00 USD
20/10/2022	Prototype SRv-6 - [Prototype]	00:36:02 21:15:00 - 21:51:02	Tsigereda Nebai 0.00 USD
20/10/2022	Prototype SRv-6 - [Prototype]	01:34:41 13:19:06 - 14:53:47	Tsigereda Nebai 0.00 USD
20/10/2022	Prototype SRv-6 - [Prototype]	00:29:39 11:41:33 - 12:11:12	Tsigereda Nebai 0.00 USD
18/10/2022	Meeting SRv-6 - [Meeting]	00:52:00 17:00:00 - 17:52:00	Tsigereda Nebai 0.00 USD
18/10/2022	Prototype SRv-6 - [Prototype]	00:30:17 16:29:43 - 17:00:00	Tsigereda Nebai 0.00 USD
18/10/2022	Meeting SRv-6 - [Meeting, Prototype]	01:05:22 15:15:00 - 16:20:22	Tsigereda Nebai 0.00 USD
18/10/2022	Prototype SRv-6 - [Prototype]	01:03:31 13:54:01 - 14:57:32	Tsigereda Nebai 0.00 USD
18/10/2022	Prototype SRv-6 - [Prototype]	00:40:59 12:57:00 - 13:37:59	Tsigereda Nebai 0.00 USD
18/10/2022	Prototype SRv-6 - [Prototype]	00:11:06 12:30:00 - 12:41:06	Tsigereda Nebai 0.00 USD
18/10/2022	SRV6-64 SRv-6 - [Documentation]	01:33:19 10:48:10 - 12:21:29	Arnaud Kenzler 0.00 USD
18/10/2022	SRV6-64 SRv-6 - [Documentation]	00:47:18 09:32:32 - 10:19:50	Arnaud Kenzler 0.00 USD
18/10/2022	SRV6-14 SRv-6 - [Requirements]	00:33:21 08:58:58 - 09:32:19	Arnaud Kenzler 0.00 USD
15/10/2022	SRV6-64 SRv-6 - [Documentation]	01:28:03 14:36:54 - 16:04:57	Arnaud Kenzler 0.00 USD

14/10/2022	Automation SRv-6 - [Prototype]	00:43:17 13:50:39 - 14:33:56	Tsigereda Nebai 0.00 USD
14/10/2022	Frontend - React SRv-6 - [Prototype]	03:21:35 09:45:00 - 13:06:35	Tsigereda Nebai 0.00 USD
13/10/2022	YAML Files SRv-6 - [Prototype]	01:33:15 21:21:29 - 22:54:44	Tsigereda Nebai 0.00 USD
13/10/2022	Frontend - React SRv-6 - [Prototype]	02:11:04 18:36:09 - 20:47:13	Tsigereda Nebai 0.00 USD
13/10/2022	Frontend - React SRv-6 - [Prototype]	00:24:20 15:04:30 - 15:28:50	Tsigereda Nebai 0.00 USD
13/10/2022	Frontend - React SRv-6 - [Prototype]	00:13:33 11:46:00 - 11:59:33	Tsigereda Nebai 0.00 USD
12/10/2022	Documentation SRv-6 - [Documentation]	02:59:55 21:50:00 - 00:49:55	Tsigereda Nebai 0.00 USD
11/10/2022	(Without Description) SRv-6 - [Meeting]	03:00:00 14:00:00 - 17:00:00	Arnaud Kenzler 0.00 USD
11/10/2022	Meeting SRv-6 - [Documentation, Project Planning, Meeting]	02:56:00 14:00:00 - 16:56:00	Tsigereda Nebai 0.00 USD
10/10/2022	Document time SRv-6 - [Documentation]	00:16:22 16:15:00 - 16:31:22	Tsigereda Nebai 0.00 USD
10/10/2022	Prototype SRv-6 - [Prototype]	02:35:17 13:30:00 - 16:05:17	Tsigereda Nebai 0.00 USD
07/10/2022	(Without Description) SRv-6 - [Prototype]	00:00:02 15:26:59 - 15:27:01	Tsigereda Nebai 0.00 USD
07/10/2022	(Without Description) SRv-6 - [Prototype]	01:15:50 15:00:00 - 16:15:50	Tsigereda Nebai 0.00 USD
07/10/2022	(Without Description) SRv-6 - [Prototype]	01:39:32 13:02:43 - 14:42:15	Tsigereda Nebai 0.00 USD
07/10/2022	(Without Description) SRv-6 - [Prototype]	01:15:47 11:15:00 - 12:30:47	Tsigereda Nebai 0.00 USD
07/10/2022	(Without Description) SRv-6 - [Prototype]	05:00:00 11:00:00 - 16:00:00	Arnaud Kenzler 0.00 USD
07/10/2022	(Without Description) SRv-6 - [Prototype]	01:23:39 09:51:24 - 11:15:03	Tsigereda Nebai 0.00 USD

05/10/2022	(Without Description) SRv-6 - [Meeting]	01:00:00 17:00:00 - 18:00:00	Arnaud Kenzler 0.00 USD
05/10/2022	Update Docu SRv-6 - [Documentation]	00:19:36 13:10:00 - 13:29:36	Tsigereda Nebai 0.00 USD
04/10/2022	Meeting SRv-6 - [Documentation, Project Planning]	02:48:07 15:06:00 - 17:54:07	Tsigereda Nebai 0.00 USD
04/10/2022	SRV6-17 SRv-6 - [Requirements]	02:22:54 14:37:06 - 17:00:00	Arnaud Kenzler 0.00 USD
03/10/2022	Domain Model Analysis SRv-6 - [Documentation, Requirements]	00:21:12 14:23:48 - 14:45:00	Tsigereda Nebai 0.00 USD
03/10/2022	Testing central-frontend SRv-6 - [Prototype]	01:30:00 09:30:00 - 11:00:00	Tsigereda Nebai 0.00 USD
30/09/2022	Testing central-frontend SRv-6 - [Prototype]	00:13:44 16:22:00 - 16:35:44	Tsigereda Nebai 0.00 USD
30/09/2022	Clean docu SRv-6 - [Documentation]	01:28:34 14:52:35 - 16:21:09	Tsigereda Nebai 0.00 USD
30/09/2022	setup SRv-6	02:10:05 14:45:33 - 16:55:38	Arnaud Kenzler 0.00 USD
30/09/2022	Testing central-frontend SRv-6 - [Prototype]	00:22:25 14:30:00 - 14:52:25	Tsigereda Nebai 0.00 USD
30/09/2022	Testing central-frontend SRv-6 - [Prototype]	01:25:27 12:49:00 - 14:14:27	Tsigereda Nebai 0.00 USD
30/09/2022	setup SRv-6	00:55:18 12:32:21 - 13:27:39	Arnaud Kenzler 0.00 USD
30/09/2022	SRV6-14 SRv-6 - [Requirements]	00:21:34 11:42:25 - 12:03:59	Arnaud Kenzler 0.00 USD
30/09/2022	Testing central-frontend SRv-6 - [Prototype]	01:12:42 11:00:57 - 12:13:39	Tsigereda Nebai 0.00 USD
30/09/2022	Testing central-frontend SRv-6 - [Prototype]	00:22:50 09:38:23 - 10:01:13	Tsigereda Nebai 0.00 USD
29/09/2022	Refine NFRs according to the feedback SRv-6 - [Documentation]	01:41:37 21:08:45 - 22:50:22	Tsigereda Nebai 0.00 USD
29/09/2022	Refine NFRs according to the feedback SRv-6 - [Documentation]	00:15:49 11:38:52 - 11:54:41	Tsigereda Nebai 0.00 USD

---

28/09/2022

SRV6-14

SRv-6 - [Requirements]

00:29:54

14:33:54 - 15:03:48

Arnaud Kenzler

0.00 USD

# Detailed report

26/10/2022 - 13/12/2022



Total: 216:02:52 Billable: 216:02:52 Amount: 0.00 USD

Date	Description	Duration	User
13/12/2022	Meeting SRv-6 - [Meeting]	00:34:58 16:57:20 - 17:32:18	Tsigereda Nebai 0.00 USD
13/12/2022	Documentation SRv-6 - [Documentation]	00:38:05 16:19:09 - 16:57:14	Tsigereda Nebai 0.00 USD
13/12/2022	Documentation, Merge SRv-6 - [Documentation, Epic Work Items]	03:08:23 13:10:37 - 16:19:00	Tsigereda Nebai 0.00 USD
13/12/2022	backend SRv-6 - [Prototype]	07:17:00 10:28:00 - 17:45:00	Arnaud Kenzler 0.00 USD
10/12/2022	Config-API SRv-6 - [Epic Work Items]	02:55:20 21:50:00 - 00:45:20	Tsigereda Nebai 0.00 USD
10/12/2022	Config-API SRv-6 - [Epic Work Items]	03:06:44 13:09:59 - 16:16:43	Tsigereda Nebai 0.00 USD
09/12/2022	Config-API SRv-6 - [Epic Work Items]	03:45:00 14:00:00 - 17:45:00	Tsigereda Nebai 0.00 USD
09/12/2022	frontend SRv-6 - [Prototype]	04:30:00 13:00:00 - 17:30:00	Arnaud Kenzler 0.00 USD
09/12/2022	Config-API SRv-6 - [Epic Work Items]	01:52:58 12:52:10 - 14:45:08	Tsigereda Nebai 0.00 USD
09/12/2022	Config-API SRv-6 - [Epic Work Items]	00:25:44 11:15:19 - 11:41:03	Tsigereda Nebai 0.00 USD
09/12/2022	Config-API SRv-6 - [Epic Work Items]	00:05:07 10:03:12 - 10:08:19	Tsigereda Nebai 0.00 USD
06/12/2022	Meeting SRv-6 - [Meeting]	02:19:23 15:57:03 - 18:16:26	Tsigereda Nebai 0.00 USD
06/12/2022	Config SRv-6 - [Epic Work Items]	01:02:00 14:55:00 - 15:57:00	Tsigereda Nebai 0.00 USD
06/12/2022	Central-Frontend integration SRv-6 - [Epic Work Items]	00:28:53 14:27:00 - 14:55:53	Tsigereda Nebai 0.00 USD

06/12/2022	Documentation SRv-6 - [Documentation]	01:16:46 13:10:14 - 14:27:00	Tsigereda Nebai 0.00 USD
06/12/2022	backend SRv-6 - [Prototype]	07:17:00 10:28:00 - 17:45:00	Arnaud Kenzler 0.00 USD
06/12/2022	Documentation SRv-6 - [Documentation]	01:50:01 10:00:07 - 11:50:08	Tsigereda Nebai 0.00 USD
02/12/2022	Central-Frontend integration SRv-6 - [Epic Work Items]	00:04:02 15:26:55 - 15:30:57	Tsigereda Nebai 0.00 USD
02/12/2022	Central-Frontend integration SRv-6 - [Epic Work Items]	02:10:27 13:00:33 - 15:11:00	Tsigereda Nebai 0.00 USD
02/12/2022	Config-API SRv-6 - [Epic Work Items]	02:36:19 09:23:00 - 11:59:19	Tsigereda Nebai 0.00 USD
29/11/2022	Meeting SRv-6 - [Meeting]	00:51:13 17:06:33 - 17:57:46	Tsigereda Nebai 0.00 USD
29/11/2022	Central-Frontend integration SRv-6 - [Epic Work Items]	01:25:18 15:32:42 - 16:58:00	Tsigereda Nebai 0.00 USD
29/11/2022	C4 Model SRv-6 - [Documentation]	00:00:07 15:32:24 - 15:32:31	Tsigereda Nebai 0.00 USD
29/11/2022	Central-Frontend integration SRv-6 - [Epic Work Items]	00:04:43 15:19:00 - 15:23:43	Tsigereda Nebai 0.00 USD
29/11/2022	Meeting SRv-6 - [Meeting]	01:01:16 14:17:25 - 15:18:41	Tsigereda Nebai 0.00 USD
29/11/2022	CICD SRv-6 - [Epic Work Items]	01:11:36 13:00:00 - 14:11:36	Tsigereda Nebai 0.00 USD
29/11/2022	CICD SRv-6 - [Epic Work Items]	00:38:23 11:17:37 - 11:56:00	Tsigereda Nebai 0.00 USD
29/11/2022	C4 Model SRv-6 - [Documentation]	01:04:42 10:12:51 - 11:17:33	Tsigereda Nebai 0.00 USD
28/11/2022	C4 Model SRv-6 - [Documentation]	02:10:04 15:24:22 - 17:34:26	Tsigereda Nebai 0.00 USD
28/11/2022	C4 Model SRv-6 - [Documentation]	01:24:53 11:17:55 - 12:42:48	Tsigereda Nebai 0.00 USD
28/11/2022	C4 Model SRv-6 - [Documentation]	00:43:08 10:21:13 - 11:04:21	Tsigereda Nebai 0.00 USD

26/11/2022	Automation SRv-6 - [Epic Work Items]	01:57:36 12:43:24 - 14:41:00	Tsigereda Nebai 0.00 USD
26/11/2022	Automation SRv-6 - [Epic Work Items]	00:03:00 12:28:00 - 12:31:00	Tsigereda Nebai 0.00 USD
26/11/2022	Automation SRv-6 - [Epic Work Items]	00:23:00 12:20:00 - 12:43:00	Tsigereda Nebai 0.00 USD
26/11/2022	Automation SRv-6 - [Epic Work Items]	00:12:31 11:51:05 - 12:03:36	Tsigereda Nebai 0.00 USD
26/11/2022	Automation SRv-6 - [Epic Work Items]	00:06:49 11:39:01 - 11:45:50	Tsigereda Nebai 0.00 USD
25/11/2022	Automation SRv-6 - [Epic Work Items]	00:26:06 16:03:54 - 16:30:00	Tsigereda Nebai 0.00 USD
25/11/2022	Meeting SRv-6 - [Meeting]	01:16:32 14:47:19 - 16:03:51	Tsigereda Nebai 0.00 USD
25/11/2022	Meeting SRv-6 - [Meeting]	01:15:00 14:45:00 - 16:00:00	Arnaud Kenzler 0.00 USD
25/11/2022	Automation SRv-6 - [Epic Work Items]	01:35:27 13:11:33 - 14:47:00	Tsigereda Nebai 0.00 USD
25/11/2022	Update Documentation SRv-6 - [Documentation]	01:15:13 11:06:15 - 12:21:28	Tsigereda Nebai 0.00 USD
25/11/2022	frontend SRv-6 - [Prototype]	03:45:00 11:00:00 - 14:45:00	Arnaud Kenzler 0.00 USD
25/11/2022	Update Documentation SRv-6 - [Documentation]	00:28:25 10:01:35 - 10:30:00	Tsigereda Nebai 0.00 USD
23/11/2022	Automation SRv-6 - [Epic Work Items]	01:37:41 16:16:19 - 17:54:00	Tsigereda Nebai 0.00 USD
23/11/2022	Automation SRv-6 - [Epic Work Items]	01:07:21 15:08:51 - 16:16:12	Tsigereda Nebai 0.00 USD
23/11/2022	Automation SRv-6 - [Epic Work Items]	00:29:57 14:30:03 - 15:00:00	Tsigereda Nebai 0.00 USD
22/11/2022	Automation SRv-6 - [Epic Work Items]	01:49:13 15:13:14 - 17:02:27	Tsigereda Nebai 0.00 USD
22/11/2022	Automation SRv-6 - [Epic Work Items]	00:46:51 14:15:54 - 15:02:45	Tsigereda Nebai 0.00 USD



22/11/2022	Automation SRv-6 - [Epic Work Items]	01:00:50 13:05:40 - 14:06:30	Tsigereda Nebai 0.00 USD
22/11/2022	frontend SRv-6 - [Prototype]	04:30:00 13:00:00 - 17:30:00	Arnaud Kenzler 0.00 USD
20/11/2022	Automation SRv-6 - [Epic Work Items]	01:39:33 13:00:38 - 14:40:11	Tsigereda Nebai 0.00 USD
19/11/2022	Automation SRv-6 - [Epic Work Items]	02:07:00 21:23:00 - 23:30:00	Tsigereda Nebai 0.00 USD
19/11/2022	Automation SRv-6 - [Epic Work Items]	02:30:00 15:00:00 - 17:30:00	Tsigereda Nebai 0.00 USD
18/11/2022	frontend SRv-6 - [Meeting]	03:30:00 14:00:00 - 17:30:00	Arnaud Kenzler 0.00 USD
18/11/2022	Automation SRv-6 - [Epic Work Items]	02:39:03 13:23:43 - 16:02:46	Tsigereda Nebai 0.00 USD
18/11/2022	frontend SRv-6 - [Prototype]	03:45:00 11:00:00 - 14:45:00	Arnaud Kenzler 0.00 USD
18/11/2022	Automation SRv-6 - [Epic Work Items]	00:48:44 10:58:53 - 11:47:37	Tsigereda Nebai 0.00 USD
15/11/2022	Automation SRv-6 - [Epic Work Items]	02:46:35 21:14:00 - 00:00:35	Tsigereda Nebai 0.00 USD
15/11/2022	Meeting SRv-6 - [Meeting]	00:28:08 17:01:52 - 17:30:00	Tsigereda Nebai 0.00 USD
15/11/2022	Automation SRv-6 - [Epic Work Items]	03:48:44 13:09:00 - 16:57:44	Tsigereda Nebai 0.00 USD
15/11/2022	frontend SRv-6 - [Meeting]	08:00:00 09:30:00 - 17:30:00	Arnaud Kenzler 0.00 USD
14/11/2022	Automation-Scrapli SRv-6 - [Epic Work Items]	02:00:19 15:11:41 - 17:12:00	Tsigereda Nebai 0.00 USD
14/11/2022	Automation - go SRv-6 - [Epic Work Items]	01:04:45 12:25:00 - 13:29:45	Tsigereda Nebai 0.00 USD
14/11/2022	Automation-Scrapli SRv-6 - [Epic Work Items]	00:53:57 11:29:10 - 12:23:07	Tsigereda Nebai 0.00 USD
14/11/2022	Automation-Scrapli SRv-6 - [Epic Work Items]	00:44:56 10:42:24 - 11:27:20	Tsigereda Nebai 0.00 USD

14/11/2022	Automation-Scrapli SRv-6 - [Epic Work Items]	01:07:25 09:30:00 - 10:37:25	Tsigereda Nebai 0.00 USD
11/11/2022	Automation-Scrapli SRv-6 - [Epic Work Items]	01:53:08 14:09:54 - 16:03:02	Tsigereda Nebai 0.00 USD
11/11/2022	Automation-Scrapli SRv-6 - [Epic Work Items]	00:42:20 13:22:46 - 14:05:06	Tsigereda Nebai 0.00 USD
11/11/2022	api SRv-6 - [Prototype]	04:15:00 13:00:00 - 17:15:00	Arnaud Kenzler 0.00 USD
11/11/2022	Automation-Scrapli SRv-6 - [Epic Work Items]	01:48:58 11:20:17 - 13:09:15	Tsigereda Nebai 0.00 USD
11/11/2022	Automation-Scrapli SRv-6 - [Epic Work Items]	01:09:03 10:04:37 - 11:13:40	Tsigereda Nebai 0.00 USD
10/11/2022	Automation-Scrapli SRv-6 - [Epic Work Items]	00:25:18 10:49:27 - 11:14:45	Tsigereda Nebai 0.00 USD
10/11/2022	Automation-Scrapli SRv-6 - [Epic Work Items]	00:23:59 10:13:19 - 10:37:18	Tsigereda Nebai 0.00 USD
09/11/2022	Meeting SRv-6 - [Meeting]	01:00:00 17:15:00 - 18:15:00	Arnaud Kenzler 0.00 USD
09/11/2022	Frontend SRv-6 - [Epic Work Items]	02:19:31 15:54:34 - 18:14:05	Tsigereda Nebai 0.00 USD
09/11/2022	api SRv-6 - [Prototype]	04:00:00 13:15:00 - 17:15:00	Arnaud Kenzler 0.00 USD
09/11/2022	Frontend SRv-6 - [Epic Work Items]	03:01:35 11:11:18 - 14:12:53	Tsigereda Nebai 0.00 USD
09/11/2022	Frontend SRv-6 - [Epic Work Items]	02:35:13 08:34:19 - 11:09:32	Tsigereda Nebai 0.00 USD
08/11/2022	Frontend SRv-6 - [Epic Work Items]	03:48:31 22:44:41 - 02:33:12	Tsigereda Nebai 0.00 USD
08/11/2022	Frontend SRv-6 - [Epic Work Items]	00:04:47 18:51:00 - 18:55:47	Tsigereda Nebai 0.00 USD
08/11/2022	Frontend SRv-6 - [Epic Work Items]	02:12:37 14:17:23 - 16:30:00	Tsigereda Nebai 0.00 USD
08/11/2022	Frontend SRv-6 - [Epic Work Items]	00:00:03 14:17:20 - 14:17:23	Tsigereda Nebai 0.00 USD

08/11/2022	Frontend SRv-6 - [Epic Work Items]	01:06:48 13:10:32 - 14:17:20	Tsigereda Nebai 0.00 USD
08/11/2022	api SRv-6 - [Prototype]	07:00:00 10:00:00 - 17:00:00	Arnaud Kenzler 0.00 USD
07/11/2022	Frontend SRv-6 - [Epic Work Items, Prototype]	02:42:55 18:10:00 - 20:52:55	Tsigereda Nebai 0.00 USD
07/11/2022	Frontend SRv-6 - [Epic Work Items, Prototype]	02:31:25 13:32:13 - 16:03:38	Tsigereda Nebai 0.00 USD
06/11/2022	Automation - normir SRv-6 - [Prototype]	01:32:04 21:50:22 - 23:22:26	Tsigereda Nebai 0.00 USD
06/11/2022	Automation - normir SRv-6 - [Prototype]	00:55:54 20:52:10 - 21:48:04	Tsigereda Nebai 0.00 USD
05/11/2022	Automation - normir SRv-6 - [Prototype]	00:23:37 22:07:27 - 22:31:04	Tsigereda Nebai 0.00 USD
05/11/2022	Automation - normir SRv-6 - [Prototype]	01:22:33 20:33:24 - 21:55:57	Tsigereda Nebai 0.00 USD
05/11/2022	Automation - normir SRv-6 - [Prototype]	02:27:22 14:00:00 - 16:27:22	Tsigereda Nebai 0.00 USD
04/11/2022	Frontend SRv-6 - [Prototype]	01:21:05 21:15:13 - 22:36:18	Tsigereda Nebai 0.00 USD
04/11/2022	api SRv-6 - [Prototype]	08:30:00 16:00:00 - 00:30:00	Arnaud Kenzler 0.00 USD
04/11/2022	Automation - go SRv-6 - [Prototype]	00:25:47 15:39:30 - 16:05:17	Tsigereda Nebai 0.00 USD
04/11/2022	Automation - go SRv-6 - [Prototype]	01:03:47 14:08:37 - 15:12:24	Tsigereda Nebai 0.00 USD
04/11/2022	Automation - go SRv-6 - [Prototype]	00:40:22 13:08:19 - 13:48:41	Tsigereda Nebai 0.00 USD
04/11/2022	Automation - go SRv-6 - [Prototype]	01:53:15 10:10:00 - 12:03:15	Tsigereda Nebai 0.00 USD
03/11/2022	Automation - go SRv-6 - [Prototype]	00:19:51 20:58:05 - 21:17:56	Tsigereda Nebai 0.00 USD
03/11/2022	Automation - go SRv-6 - [Prototype]	00:25:46 20:30:00 - 20:55:46	Tsigereda Nebai 0.00 USD

03/11/2022	Automation - go SRv-6 - [Prototype]	01:20:00 15:05:00 - 16:25:00	Tsigereda Nebai 0.00 USD
02/11/2022	Automation - go SRv-6 - [Prototype]	02:35:00 14:20:00 - 16:55:00	Tsigereda Nebai 0.00 USD
01/11/2022	Automation SRv-6 - [Prototype]	01:53:57 13:57:30 - 15:51:27	Tsigereda Nebai 0.00 USD
01/11/2022	Automation SRv-6 - [Prototype]	01:38:22 09:47:11 - 11:25:33	Tsigereda Nebai 0.00 USD
01/11/2022	docker SRv-6 - [Prototype]	08:30:00 07:30:00 - 16:00:00	Arnaud Kenzler 0.00 USD
30/10/2022	Automation SRv-6 - [Prototype]	04:12:05 12:50:00 - 17:02:05	Tsigereda Nebai 0.00 USD
28/10/2022	api SRv-6 - [Prototype]	07:30:00 16:40:00 - 00:10:00	Arnaud Kenzler 0.00 USD
28/10/2022	Automation SRv-6 - [Prototype]	00:46:52 15:14:29 - 16:01:21	Tsigereda Nebai 0.00 USD
28/10/2022	Automation SRv-6 - [Prototype]	01:00:00 14:00:00 - 15:00:00	Tsigereda Nebai 0.00 USD
28/10/2022	Prototype SRv-6 - [Prototype]	01:02:31 12:46:57 - 13:49:28	Tsigereda Nebai 0.00 USD
28/10/2022	Prototype SRv-6 - [Prototype]	01:11:17 10:52:55 - 12:04:12	Tsigereda Nebai 0.00 USD

# Detailed report

14/12/2022 - 23/12/2022



Total: 57:25:36 Billable: 57:25:36 Amount: 0.00 USD

Date	Description	Duration	User
23/12/2022	Documentation SRv-6 - [Documentation]	01:04:44 09:25:16 - 10:30:00	Tsigereda Nebai 0.00 USD
23/12/2022	docu SRv-6 - [Documentation]	04:30:00 07:30:00 - 12:00:00	Arnaud Kenzler 0.00 USD
22/12/2022	Proof read documentation SRv-6 - [Documentation]	00:32:29 21:14:44 - 21:47:13	Tsigereda Nebai 0.00 USD
22/12/2022	Proof read documentation SRv-6 - [Documentation]	00:34:04 20:15:26 - 20:49:30	Tsigereda Nebai 0.00 USD
22/12/2022	Documentation SRv-6 - [Documentation]	02:19:29 14:13:37 - 16:33:06	Tsigereda Nebai 0.00 USD
22/12/2022	Documentation SRv-6 - [Documentation]	00:02:01 13:53:38 - 13:55:39	Tsigereda Nebai 0.00 USD
22/12/2022	Documentation SRv-6 - [Documentation]	00:30:04 11:12:22 - 11:42:26	Tsigereda Nebai 0.00 USD
22/12/2022	Documentation and Meeting SRv-6 - [Documentation, Meeting]	00:34:48 10:37:32 - 11:12:20	Tsigereda Nebai 0.00 USD
22/12/2022	docu SRv-6 - [Documentation]	07:00:00 10:00:00 - 17:00:00	Arnaud Kenzler 0.00 USD
22/12/2022	Documentation SRv-6 - [Documentation]	00:49:53 09:41:36 - 10:31:29	Tsigereda Nebai 0.00 USD
22/12/2022	Documentation SRv-6 - [Documentation]	00:30:06 09:01:29 - 09:31:35	Tsigereda Nebai 0.00 USD
22/12/2022	Documentation SRv-6 - [Documentation]	00:07:51 08:03:35 - 08:11:26	Tsigereda Nebai 0.00 USD
21/12/2022	Documentation SRv-6 - [Documentation]	01:16:05 12:07:45 - 13:23:50	Tsigereda Nebai 0.00 USD
21/12/2022	Documentation SRv-6 - [Documentation]	00:29:35 10:39:00 - 11:08:35	Tsigereda Nebai 0.00 USD

21/12/2022	Documentation SRv-6 - [Documentation]	02:09:56 08:30:00 - 10:39:56	Tsigereda Nebai 0.00 USD
20/12/2022	Documentation SRv-6 - [Documentation]	01:59:26 16:12:01 - 18:11:27	Tsigereda Nebai 0.00 USD
20/12/2022	Documentation SRv-6 - [Documentation]	02:40:27 13:24:43 - 16:05:10	Tsigereda Nebai 0.00 USD
20/12/2022	docu SRv-6 - [Documentation]	05:30:00 13:00:00 - 18:30:00	Arnaud Kenzler 0.00 USD
20/12/2022	Documentation SRv-6 - [Documentation]	00:35:40 11:05:18 - 11:40:58	Tsigereda Nebai 0.00 USD
20/12/2022	frontend SRv-6 - [Prototype]	02:02:16 10:00:00 - 12:02:16	Arnaud Kenzler 0.00 USD
20/12/2022	Documentation SRv-6 - [Documentation]	00:56:21 10:00:00 - 10:56:21	Tsigereda Nebai 0.00 USD
20/12/2022	Documentation SRv-6 - [Documentation]	00:24:00 07:45:00 - 08:09:00	Tsigereda Nebai 0.00 USD
19/12/2022	frontend SRv-6 - [Prototype]	02:30:00 16:00:00 - 18:30:00	Arnaud Kenzler 0.00 USD
18/12/2022	docu SRv-6 - [Documentation]	05:30:00 13:00:00 - 18:30:00	Arnaud Kenzler 0.00 USD
18/12/2022	Documentation SRv-6 - [Documentation]	01:18:54 00:54:06 - 02:13:00	Tsigereda Nebai 0.00 USD
17/12/2022	backend SRv-6 - [Prototype]	03:00:00 14:00:00 - 17:00:00	Arnaud Kenzler 0.00 USD
17/12/2022	(Without Description) SRv-6 - [Documentation]	00:15:00 00:00:00 - 00:15:00	Tsigereda Nebai 0.00 USD
16/12/2022	Frontend - Home page SRv-6 - [Epic Work Items]	01:51:20 14:00:00 - 15:51:20	Tsigereda Nebai 0.00 USD
16/12/2022	backend SRv-6 - [Prototype]	04:00:00 13:00:00 - 17:00:00	Arnaud Kenzler 0.00 USD
16/12/2022	(Without Description) SRv-6 - [Epic Work Items]	00:08:26 12:49:39 - 12:58:05	Tsigereda Nebai 0.00 USD
16/12/2022	(Without Description) SRv-6 - [Epic Work Items]	00:01:08 12:40:46 - 12:41:54	Tsigereda Nebai 0.00 USD

---

16/12/2022	(Without Description) SRv-6 - [Documentation, Epic Work Items]	00:44:00 11:15:00 - 11:59:00	Tsigereda Nebai 0.00 USD
16/12/2022	(Without Description) SRv-6 - [Epic Work Items]	01:27:33 10:37:30 - 12:05:03	Tsigereda Nebai 0.00 USD

---

## 9.4 Administration



## Vereinbarung über Urheber- und Nutzungsrechte

### 1. Vereinbarung

#### 1.1.1 1. Gegenstand der Vereinbarung

Mit dieser Vereinbarung werden die Rechte über die Verwendung und die Weiterentwicklung der Ergebnisse der Studienarbeit Segment Routing v6 Network Programming (SRv6 Network Programming) von Arnaud Kenzler und Tsigereda Nebai Kidane unter der Betreuung von Prof. Laurent Metzger geregelt.

#### 1.1.2 2. Urheberrecht

Die Urheberrechte stehen der Studentin / dem Studenten zu.

#### 1.1.3 3. Verwendung

Die Ergebnisse der Arbeit dürfen sowohl von der Studentin / dem Studenten, von der OST wie von Cisco Systems Belgium nach Abschluss der Arbeit verwendet und weiterentwickelt werden

#### 1.1.4 Beilage/n:

-

Rapperswil, den 22. Dez. 2022

  
.....  
Die Studentin/der Student

Rapperswil, den.....


.....  
Der Betreuer / die Betreuerin der  
Studienarbeit

## 2. Vereinbarung

Ohne anderslautende Vereinbarungen stehen die Schutzrechte und das Know-how an der Studienarbeit oder Bachelorarbeit (nachfolgend ‚Arbeit‘ genannt) und an der in diesem Rahmen geschaffenen Güter, wie Software, sowohl dem Rechtsträger der OST Ostschweizer Fachhochschule, dem für die Arbeit verantwortlichen Professoren sowie dem Verfasser der Arbeit resp. Entwickler der in diesem Rahmen geschaffenen Güter, wie Software, zu.

Die genannten Parteien übertragen sich gegenseitig nicht exklusiv, jedoch unentgeltlich, weltweit, sachlich und zeitlich unbeschränkt die jeweiligen Schutzrechte und das Know-how an der Arbeit und an der in diesem Rahmen geschaffenen Güter, wie Software, einschliesslich dem Recht zur Weiterübertragung, ab. Entsprechend steht es jeder Partei zu, sämtliche Schutzrechte an der Arbeit resp. an der in diesem Rahmen geschaffenen Güter, wie Software, beliebig weltweit, zeitlich und sachlich unbeschränkt zu verwerten. Darunter fällt namentlich aber nicht abschliessend das Recht zur Lizenzierung in jeder Art, Umfang und Form, das Recht zur Bearbeitung und damit zur Nutzung z. B. der Software oder Komponenten hiervon als Grundlage eines neuen schutzfähigen Guts. Die Parteien erklären sich gegenseitig den Verzicht auf Namensnennung bei der Verwertung der Schutzrechte und des Know-how durch eine oder mehrere Parteien gemeinsam und stimmen namentlich zu, dass jede Partei allein unter ihrem eigenen Namen die Schutzrechte resp. das Know-how verwertet. Die vorliegende gegenseitige unentgeltliche Übertragung der Schutzrechte resp. des Know-how bezieht sich auch auf Verwertungsarten, welche heute noch nicht bekannt sind.

Rapperswil, den 22. Dez. 2022

  
.....  
Die Studentin/der Student

Rapperswil, den.....

.....  
Der Betreuer / die Betreuerin der  
Studienarbeit

## Einverständniserklärung Publikation auf eprints.ost.ch

- SA
- BA

Titel der Arbeit: Segment Routing v6 Network Programming (SRv6 Network Programming)

Team: Arnaud Kenzler und Tsigereda Nebai Kidane

Betreuer: Prof. Laurent Metzger

Wir sind mit der Publikation unserer Arbeit auf eprints.ost.ch einverstanden, sofern für diese Arbeit keine Geheimhaltungsvereinbarung unterzeichnet wurde.

Nach Bekanntgabe der Note haben wir die Möglichkeit innert 14 Tagen Einsprache zu erheben und das Einverständnis zur Publikation der Arbeit auf eprints.ost.ch zurückzuziehen. In diesem Falle wird nur der Abstract publiziert.

Rapperswil, 22. Dezember 2022

Name(n)

Unterschrift(en)

Arnaud Kenzler



Tsigereda Nebai Kidane



## Eigenständigkeitserklärung

### Erklärung

Ich erkläre hiermit,

- dass ich die vorliegende Arbeit selbst und ohne fremde Hilfe durchgeführt habe, ausser derjenigen, welche explizit in der Aufgabenstellung erwähnt ist oder mit der Betreuerin / dem Betreuer schriftlich vereinbart wurde,
- dass ich sämtliche verwendeten Quellen erwähnt und gemäss gängigen wissenschaftlichen Zitierregeln korrekt angegeben habe,
- dass ich keine durch Copyright geschützten Materialien (z.B. Bilder) in dieser Arbeit in unerlaubter Weise genutzt habe.

Ort, Datum: Rapperswil, 22. Dezember 2022

Name, Unterschrift:



Arnaud Kenzler



Tsigereda Nebai Kidane