BLEKINGE INSTITUTE OF TECHNOLOGY

SOFTWARE REQUIREMENT SPECIFICATION

TCP Evaluation in Semi-Live Streams

“Team Enigma”

Anirudh Kodaru

Hemanth Kumar Ravuri

Nandini Chowdary Godavarthi

Naren Naga Pavan Prithvi Tanneedi

Reventh Thiruvallur Vangeepuram

Sasidhar Podapati

Sathvik Katam

Srinand Kona

SriKavya Chavali

Vaibhav Bajaj

Venkata Sathya Sita J S Ravu

Version 1.5
Publication Date: 2015/08/24

1. PREFACE:

The Software Requirement Specification document of this project provides a description of the system architecture and the requirements.

The first edition is v1.0. The current edition is v1.4.

The document gives a description of the system architecture of the tool being developed. It provides an overview of the user requirements and the system requirements. It also shows the references for the literature review for SRS of the project.

Release v1.5 on 2015-08-24

- In section 4.1 of the document, user functional requirement REQ_USRF_6 is added. Hence the followed requirement string numbers are changed accordingly.

Release v1.4 on 2015-06-01

- In section 3 of the document the fig 3.1: system architecture is updated and a descriptive point is added in the description.
- In section 3.1, fig 3.2: design of frontend is updated and its respective description is updated.
- In section 3.2, fig 3.2: design of database is updated and its respective description is updated.
- In section 3.3, fig 3.4: design of the backend is updated and its respective description is updated.
- In section 3.4, fig 3.5: design of the RESTful API is updated and its respective description is updated.
- In section 4.1, user functional requirements are associated with the tests mentioned in the acceptance test plan v1.2.
- In section 4.1, three user functional requirements namely data import, cross correlation analysis and add/remove streams are removed.
- In section 4.1, stream selection user requirement is changed to consumer login on the dashboard and its description is updated.
- In section 4.1, scalability a user non-functional requirement description is upadated.

Release v1.3 on 2015-05-20

- In all the sections of the document figure numbers and table numbers along with their captions are provided for the respective figures and tables.
- In section 4.1, a user functional requirement namely threshold notifications is added.

Release v1.2 on 2015-05-14

- In section 3, system architecture figure description is changed. ‘S’ is referred to as switch ports.
- In section 3.1, frontend module has been updated. User authentication is added. Regarding threshold notifications, an SMS option has been added.
- In section 4.1, user requirements section has been updated. Login authentication requirement is added. Cross correlation and data import using RESTful API requirements have been separated. Simple dashboard requirement has been split into three different requirements namely add/remove stream, selection of stream and metric selection.

Release v1.1 on 2015-05-05

- Updated high level design of the tool in system architecture.
- Increased number of modules depending on the high level design of the system architecture.
- The detailed explanation of each module is updated as the high level design is updated.
- REQ_USRF_1 is divided into two different requirements.
- Two more user functional requirements are added regarding RESTful API.
- A simple dashboard is updated as a user functional requirement in place of web interface.
- Programming languages is removed from the system functional requirements.
- Compatibility is removed from the system non-functional requirements.

Release v1.0 on 2015-04-27

- Initial Release

2. GLOSSARY AND ABBREVATIONS:

SRS – Software Requirement Specification

RRD – Round Robin Database

The data in RRD is mainly a time series data like network bandwidth, CPU load etc., which is stored in circular buffers.

DPMI – Distributed Passive Measurement Infrastructure

This structure allows for an efficient use of passive monitoring equipment in order to supply researchers and network managers with up-to-date and relevant data. [1]

API – Application Programming Interface

A set of routines, protocols and tools for building software applications.

RTT – Round Trip Time

The total time taken for a packet to be sent from source to destination and the acknowledgement received from destination to source.

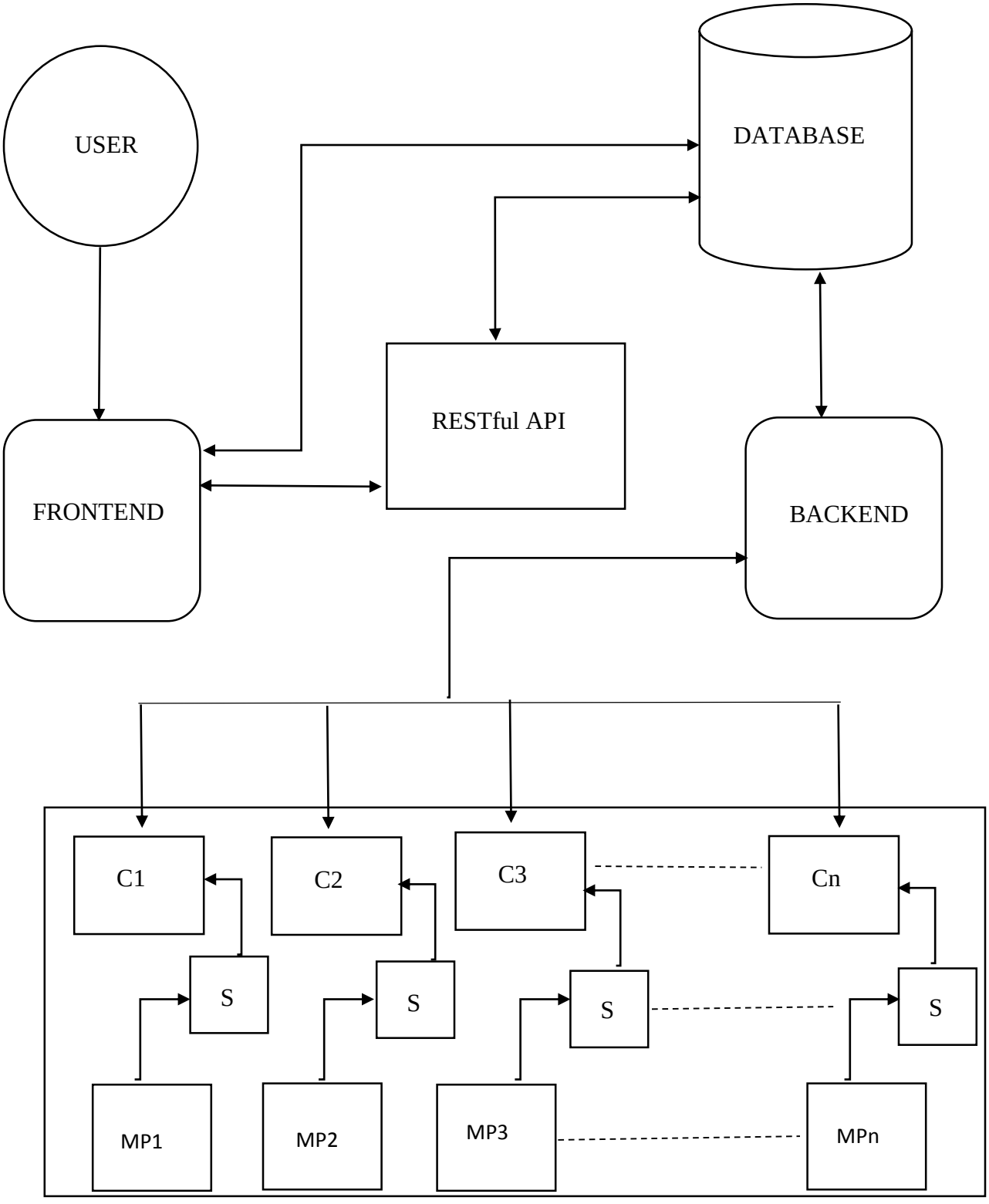
GUI – Graphical User Interface

Allows the user to interact with electronic devices through graphical icons and audio-visual indicators.

TCP – Transmission Control Protocol

Protocol used to exchange data between two communicating hosts through an established connection.

3. SYSTEM ARCHITECTURE:



C1, C2 – Consumers in DPMI

S - Switch ports to broadcast the data packets

MP1, MP2 - Measuring points in DPMI

Fig 3.1: System Architecture

- The Fig 3.1 shows the system architecture and shows the working of the software system.
- The high level design architecture is divided into modules such as:
    - Module 1: Frontend
    - Module 2: Database
    - Module 3: Backend
    - Module 4: RESTful API
- The modules show the different parts of the high level diagram and explain the working of the tool.
- Fig 3.1 shows the system architecture of the tool. The multiple measuring points, multiple switch ports and multiple consumers in the DPMI show the functionality of the DPMI. The tool works on a single consumer at a time.

3.1 Module 1: Frontend:

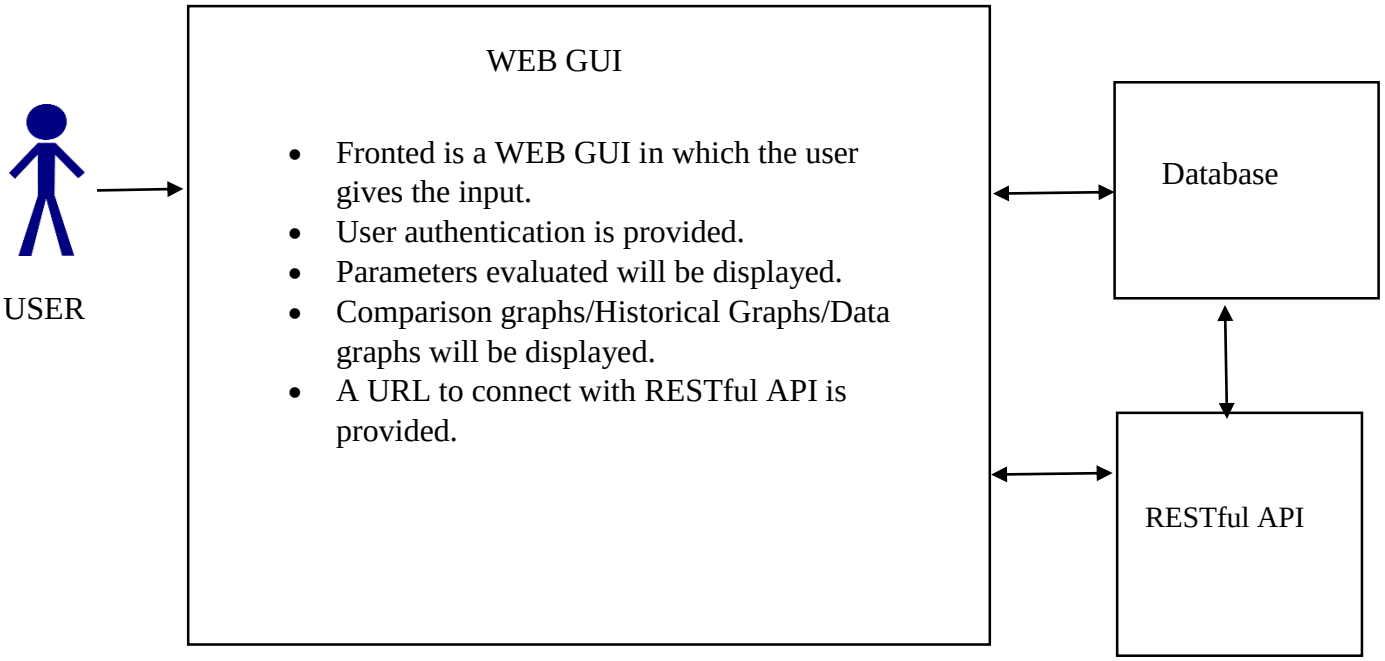


Fig 3.2: Design of Frontend

- Frontend is the part where the code is written to create a WEB GUI, where the activity of the tool will be observed.
- Firstly, the user provides authentication details to gain access of the tool.
- The user gives the input to the tool on the webpage. One of the inputs will be the address of the consumer where the data packets will be collected from.
- The input will give an option of selecting the metrics required.

- The total input given on the webpage is saved in the database (MySQL tool). The data is sent to the database through API.
- The webpage will also display the evaluated parameters from database of the tool.
- The evaluated parameters are stored in the database (RRD tool) by the backend of the tool. The parameters are sent to the webpage as requested by the user through the API.
- The threshold levels are defined by the user on the webpage. For each metric a different threshold level is defined by the user. Also, a slot for providing an email address is provided on the webpage.
- Comparison graphs/Historical graphs/Data graphs (time series) are displayed on the webpage as requested by the user.

3.2 Module 2: Database:

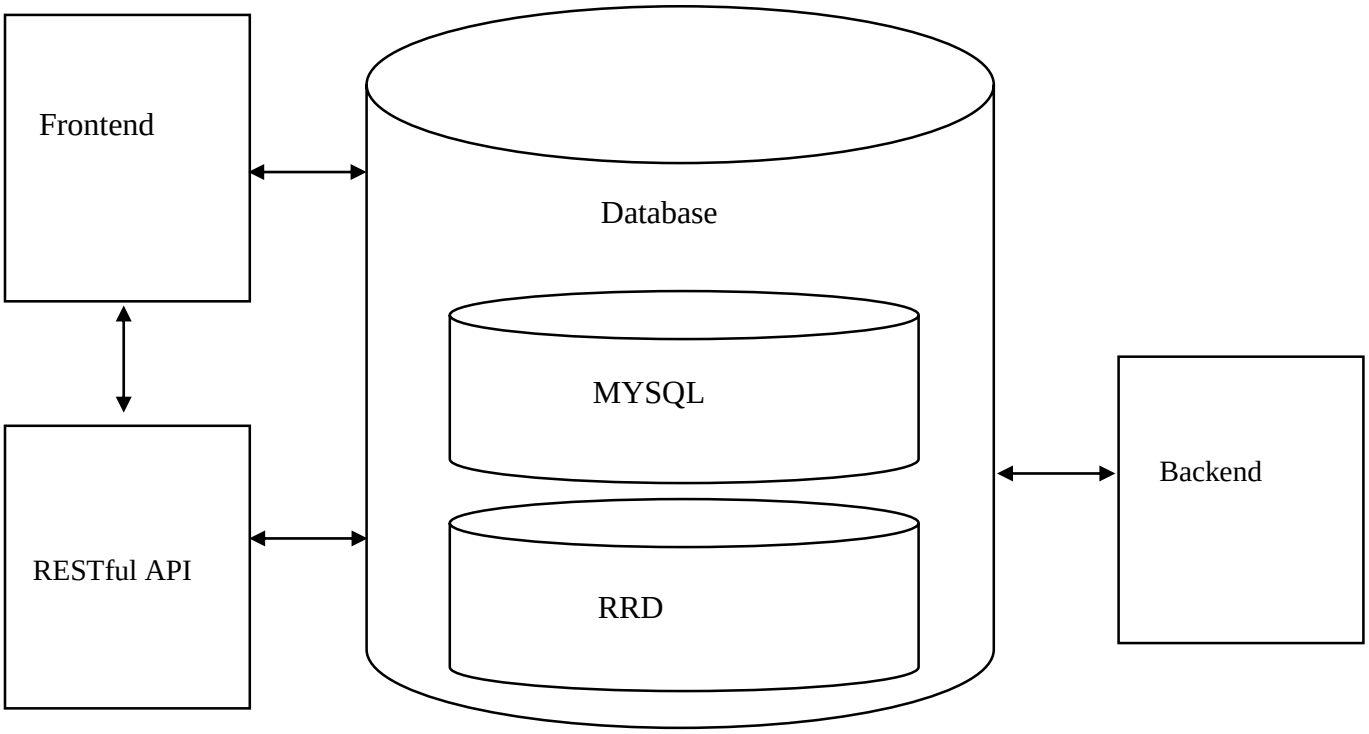


Fig 3.3: Design of Databases

- The database is used to store data exchanged during the operation of the tool. There are two types of data namely time series data and tabular data that will be stored during the operation of the tool.
- The MySQL database is used to store tabular data and the RRD is used to store the time series data which is in turn used for plotting graphs.

- Any input from the frontend is stored in the MySQL database and is then sent to the backend. This communication is carried through API.
- Evaluated data from the backend is continuously updated in the RRD as it is a time series data and is then displayed on the frontend. This communication is also carried through API.
- A RESTful API is connected to the database to have communication with any available third party.

3.3 Module 3: Backend:

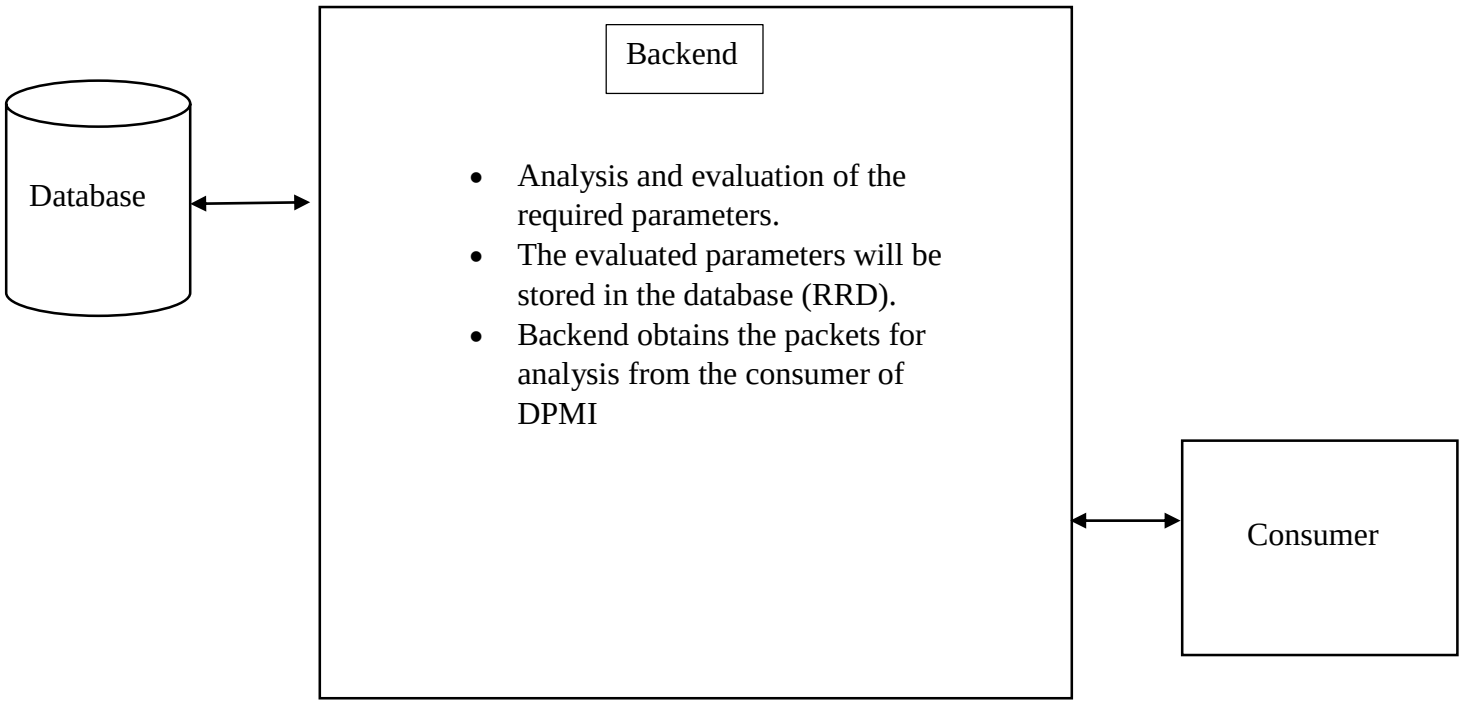


Fig 3.4: Design of Backend

- Backend receives the consumer login information from the database (MySQL). Then it receives the packets from the consumer of DPMI.
- Then the analysis of the packets will be carried out by the backend and the required parameters will be evaluated.
- The evaluated parameters will then be stored in the database (RRD) as it is a time series data.
- The backend must be able to generate notifications on the basis of the threshold levels defined by the user. Also, the generated notifications must be controlled dynamically (emails and traps).

3.4: Module 4: RESTful API:

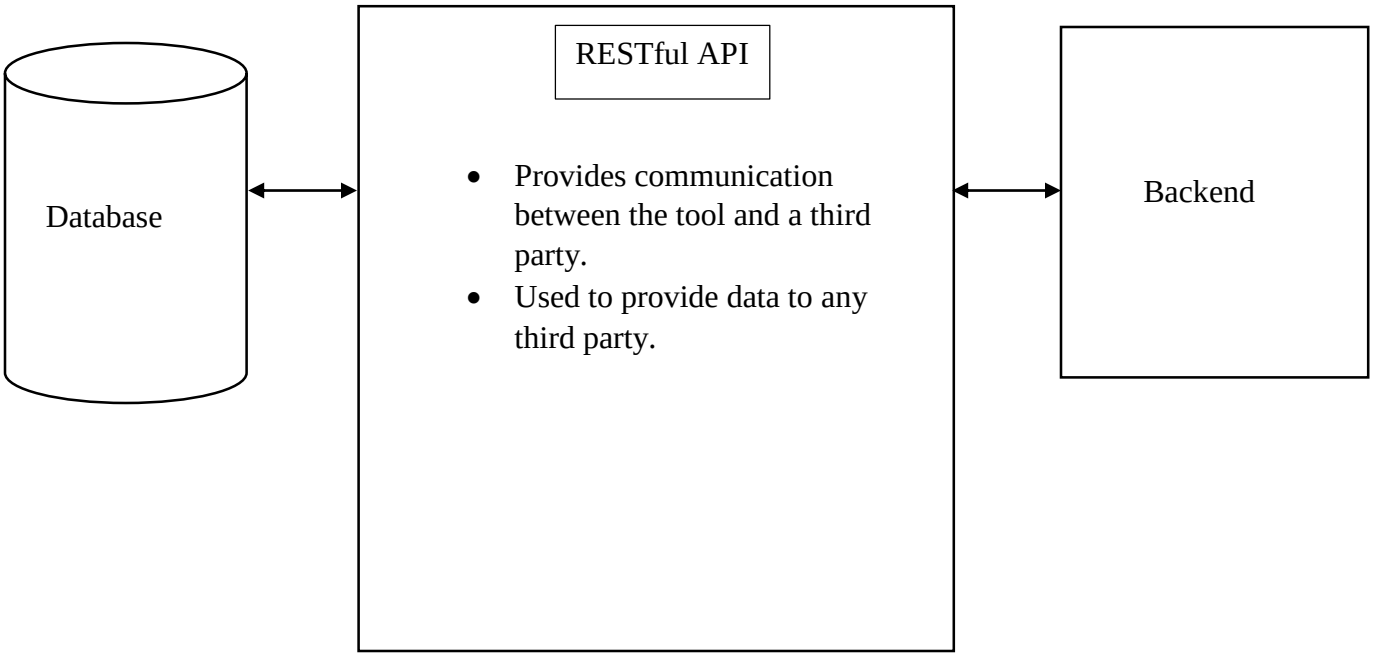


Fig 3.5: Design of RESTful API

- RESTful API is used to establish a communication between the tool developed and any third party data source.
- A RESTful interface of web service is designed with PHP script. It facilitates accessibility of data for a 3rd party through HTTP GET request and retrieves/store data from MySQL database, shows service name, status and uptime in JSON format.
- RESTful API will be used to provide evaluated data to any third party data source when requested in a specified format (JSON format).
- A URL is provided to the third party for accessing the data that is obtained through the RESTFUL API.
- The third party will be opening the URL in the browser and will see the data in JSON format.

4. REQUIREMENTS:

4.1. User Requirements:

User Functional Requirements:

REQ_USRF_1: Login Authentication

Requirement ID	REQ_USRF_1
Creation Date	2015-05-14
Change Date	
Module	Frontend
Type	Functional
Dependencies	
Test	ATP_1
Assignee	
Description	The user needs to provide authentication to have access to the tool.
Comment	

Table 4.1: REQ_USRF_1: Login Authentication

REQ_USRF_2: Capturing of TCP Packets

Requirement ID	REQ_USRF_2
Creation Date	2015-04-27
Change Date	2015-05-14
Module	Backend
Type	Functional
Dependencies	
Test	ATP_5
Assignee	
Description	The TCP traffic from the streams at consumer are captured using a backend script.
Comment	

Table 4.2: REQ_USRF_2: Capturing of TCP packets

REQ_USRF_3: Analysis of TCP Packets

Requirement ID	REQ_USRF_3
Creation Date	2015-04-27
Change Date	2015-05-14
Module	Backend, Database
Type	Functional
Dependencies	
Test	ATP_6
Assignee	
Description	The captured TCP packets will be analyzed at the backend. The parameters such as RTT, Socket-setup time and Data rate per stream will be evaluated and stored in the database.
Comment	

Table 4.3: REQ_USRF_3: Analysis of TCP Packets

REQ_USRF_4: Generation of Graphs

Requirement ID	REQ_USRF_4
Creation Date	2015-04-27
Change Date	2015-05-14
Module	Database, Frontend
Type	Functional
Dependencies	
Test	ATP_4
Assignee	
Description	The evaluated parameters stored in the database are retrieved by the frontend. The parameters such as RTT, Socket-setup time and Data rate per stream and their aggregate with respect to time will be shown in a visual graphical format.
Comment	The RRD will produce the graphs with self-defined timescales. The timescales are not selectable by the user.

Table 4.4: REQ_USRF_4: Generation of Graphs

REQ_USRF_5: Data export through RESTful API

Requirement ID	REQ_USRF_5
Creation Date	2015-05-05
Change Date	2015-05-14
Module	RESTful API, Database
Type	Functional
Dependencies	
Test	ATP_9
Assignee	
Description	The tool will be able to export the evaluated data saved in the database to any third party if requested.
Comment	

Table 4.5: REQ_USRF_5: Data export through RESTful API

REQ_USRF_6: Data import through RESTful API

Requirement ID	REQ_USRF_6
Creation Date	2015-08-24
Change Date	
Module	RESTful API, Database
Type	Functional
Dependencies	
Test	ATP_9
Assignee	
Description	The tool will be able to import data from any third party in JSON format and save in the database of the tool if requested.
Comment	

Table 4.6: REQ_USRF_6: Data import through RESTful API

REQ_USRF_7: Metric selection on dashboard

Requirement ID	REQ_USRF_7
Creation Date	2015-04-27
Change Date	2015-05-14
Module	Frontend
Type	Functional
Dependencies	
Test	ATP_2
Assignee	
Description	The dashboard will have an option for metric selection which displays the requested data when selected.
Comment	

Table 4.7: REQ_USRF_7: Metric Selection on dashboard

REQ_USRF_8: Consumer login on the dashboard

Requirement ID	REQ_USRF_8
Creation Date	2015-04-27
Change Date	2015-05-14
Module	Frontend
Type	Functional
Dependencies	
Test	ATP_1
Assignee	
Description	The user will be asked to enter the consumer details on the dashboard.
Comment	

Table 4.8: REQ_USRF_8: Stream Selection on dashboard

REQ_USRF_9: Threshold notifications

Requirement ID	REQ_USRF_9
Creation Date	2015-05-20
Change Date	
Module	Frontend, Database, Backend
Type	Functional
Dependencies	
Test	ATP_7, ATP_8
Assignee	
Description	Threshold notifications are generated depending on the threshold levels defined by the user. The notifications are notified to the user through e-mail and traps.
Comment	

Table 4.9: REQ_USRF_9: Threshold notifications

User Non-Functional Requirements:

REQ_USRN_1: Scalability

Requirement ID	REQ_USRN_1
Creation Date	2015-04-27
Change Date	
Module	Backend, Database, Frontend
Type	Non-Functional
Dependencies	
Test	
Assignee	
Description	The tool will be able to analyze data from different consumers and evaluate them.
Comment	

Table 4.10: REQ_USRN_1: Scalability

REQ_USRN_2: Documentation

Requirement ID	REQ_USRN_2
Creation Date	2015-04-27
Change Date	
Module	Frontend, Database, Backend, RESTful API
Type	Non-Functional
Dependencies	
Test	
Assignee	
Description	Documentation will be provided to the client after the tool is developed and released. This will be helpful for the client in handling of the tool. Support, services and updates will be available for user in the future (if user needs any development).
Comment	

Table 4.11: REQ_USRN_2: Documentation

4.2: SYSTEM REQUIREMENTS:

System Functional Requirements:

REQ_SYSF_1: Operating System

Requirement ID	REQ_SYSF_1
Creation Date	2015-04-27
Change Date	
Module	Frontend
Type	Functional
Dependencies	
Test	
Assignee	
Description	As the requirements of the tool developed, the operating system required is UBUNTU 14.04 version with the latest version of Firefox.
Comment	

Table 4.12: REQ_SYSF_1: Operating System

REQ_SYSF_2: Database

Requirement ID	REQ_SYSF_2
Creation Date	2015-04-27
Change Date	
Module	Database, Frontend, Backend, RESTful API
Type	Functional
Dependencies	
Test	
Assignee	
Description	Databases like MySQL and RRD will be used to store and access data during the operation of the tool.
Comment	

Table 4.13: REQ_SYSF_2: Database

System Non-Functional Requirements

REQ_SYSN_1: Performance

Requirement ID	REQ_SYSN_1
Creation Date	2015-04-27
Change Date	
Module	Frontend, Backend, Database, RESTful API
Type	Non-Functional
Dependencies	
Test	
Assignee	
Description	The performance of the developed tool will be able to cope up with the semi-real time.
Comment	

Table 4.14: REQ_SYSN_1: Performance

REQ_SYSN_2: Quality of Service

Requirement ID	REQ_SYSN_2
Creation Date	2015-04-27
Change Date	
Module	Frontend, Backend, Database, RESTful API
Type	Non-Functional
Dependencies	
Test	
Assignee	
Description	The developed software system will be able to process the data accurately with minimum time required.
Comment	

Table 4.15: REQ_SYSN_2: Quality of Service

5. REFERENCES:

[1] Patrik Arlos, Markus Fiedler, and Arne A. Nilsson. *A Distributed Passive Measurement Infrastructure*, In Passive and Active Measurement Workshop (PAM05), US, 2005.

[2] Ian Sommerville, *SOFTWARE ENGINEERING*, 9th ed. Pearson Publications, 2011.