**OVERLAY NETWORK MONITORING SYSTEM**

SOFTWARE REQUIREMENTS SPECIFICATIONS

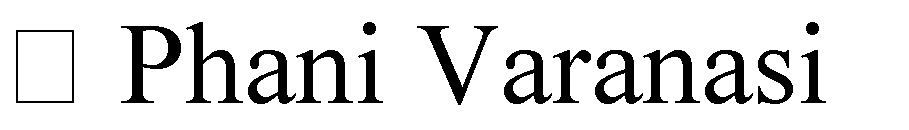
Version 1.2

**Team Name:** Smart Developers

**Team Members:**











Sana











1. **Preface:**

The main concern of the project is to provide the customer a simple and unified way of maintaining and updating its applications which interacts with the monitoring system through a common RESTful API. This is the initial version of the document.

1. Release version 1.2 on 2017-06-16:

* Made changes in User Requirements
* Updated minor errors

1. Release version 1.1 on 2016-12-18:

* Made changes in the section glossary and abbreviations
* User requirements and system requirements have been updated

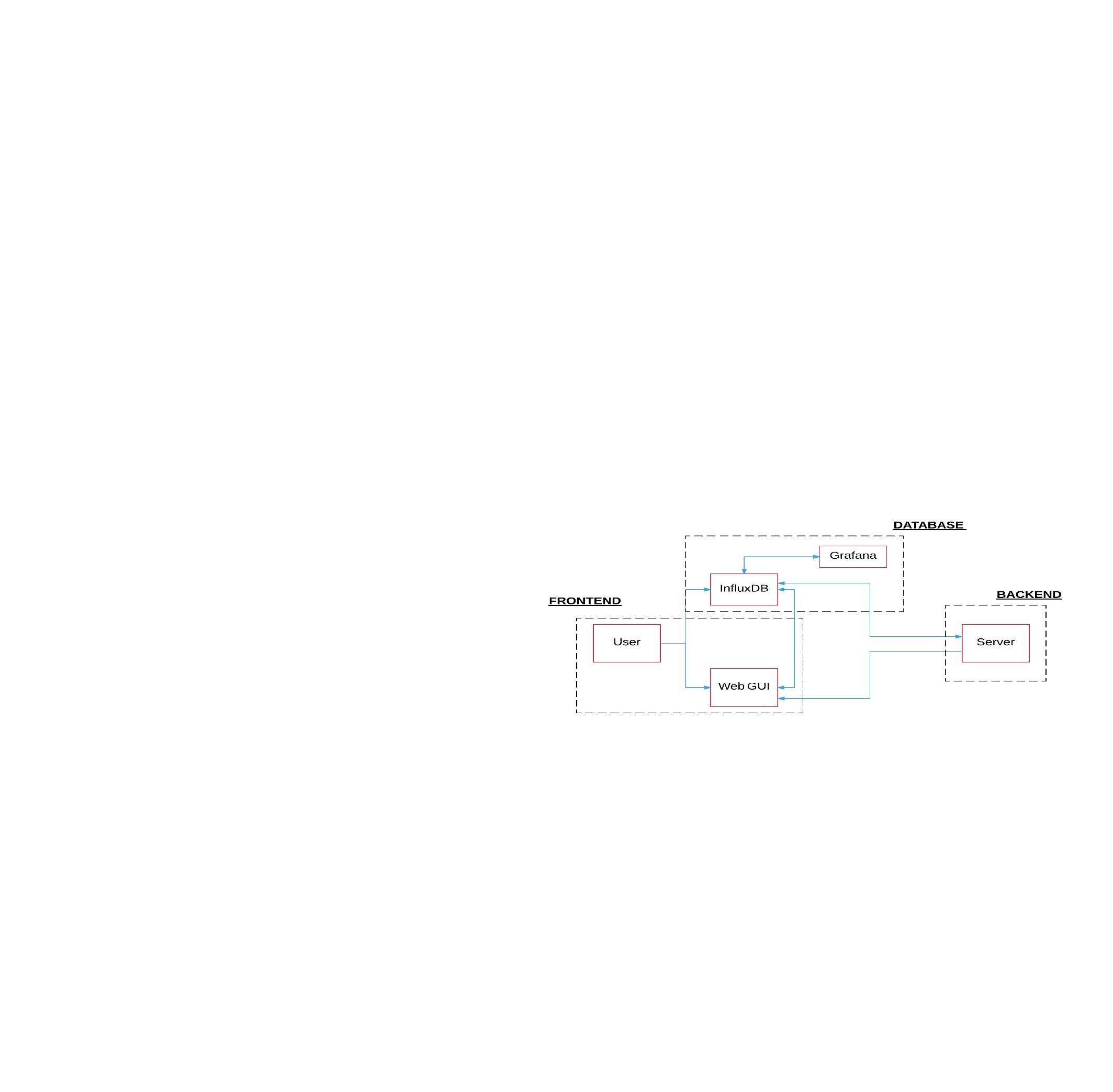
1. Release version 1.0 on 2016-04-24:

In the remainder of the document, Section II describes briefly about the basic abbreviations used in this document. Section III defines the System Architecture which is divided into Frontend, Database and Backend modules. Lastly Section IV gives the Requirements (User and System)

1. **GLOSSARY AND ABBREVATIONS:**

* **API**: Application Programming Interface
  + An API is a set of routines, protocols, and tools for building software applications.
* **InfluxDB**: Influx Database
  + InfluxDB is an open source time series database. InfluxDB has no external dependencies and provides an SQL-like language with built in time-centric functions for querying a data structure composed of measurements, series, and points.
* **Grafana**:
  + Grafana is most commonly used for visualizing time series data for Internet infrastructure and application analytics but many use it in other domains including industrial sensors, home automation, weather, and process control. Grafana features pluggable panels and data sources allowing easy extensibility and a variety of panels, including fully featured graph panels with rich visualization options. There is built in support for many of the most popular time series data sources.
* **SSL**: Secure Sockets Layer
  + SSL is a standard security technology for establishing an encrypted link between a web server and a browser.
* **RTT**: Round Trip Time
* **HTTPS**: Hyper Text Transport Protocol Secure
* **RESTful**: Representative State Transfer

1. **SYSTEM ARCHITECTURE:**



**Fig: System Architecture**

This section provides a high level view of the system architecture and is divided into three subsections. Each subsection describes a major component of the system being developed. The three main modules in the system are Frontend, Database and Backend.

**3.1) FRONTEND:**

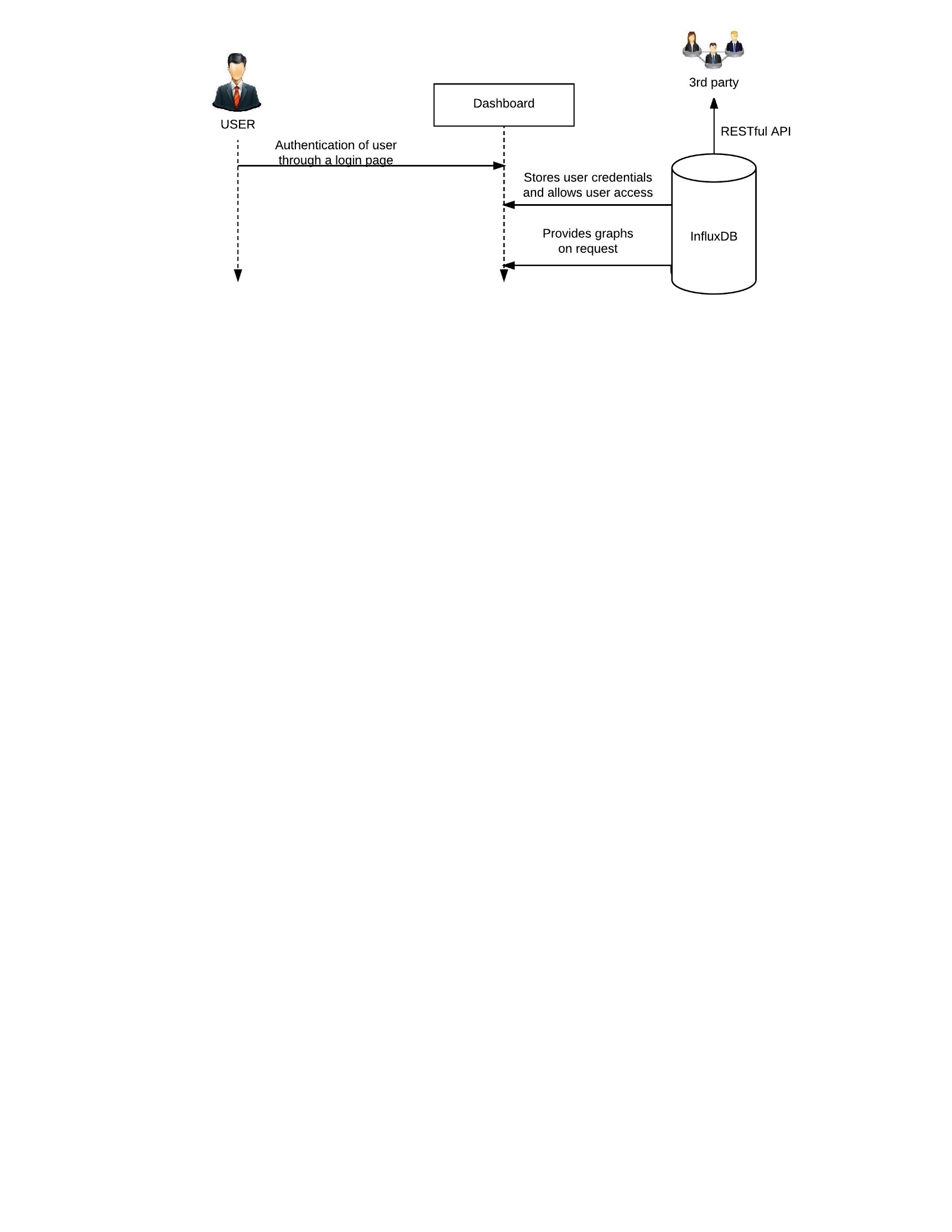


Fig: Frontend

This module represents frontend of the system architecture. The customer can check status of Overlay network services via web interface. The customer is given access to the dashboard after authentication through a login webpage.

The user can select three options: Logout, Edit account, View network data. Logout option is used to prevent unauthorized users, after current user leaves system. If user wants to edit his/her account details, edit option is used. If view network data option is chosen, user can monitor datatypes.

Graphs are generated by Perl script using data already inserted to MySQL database. These graphs will update automatically when new data arrives.

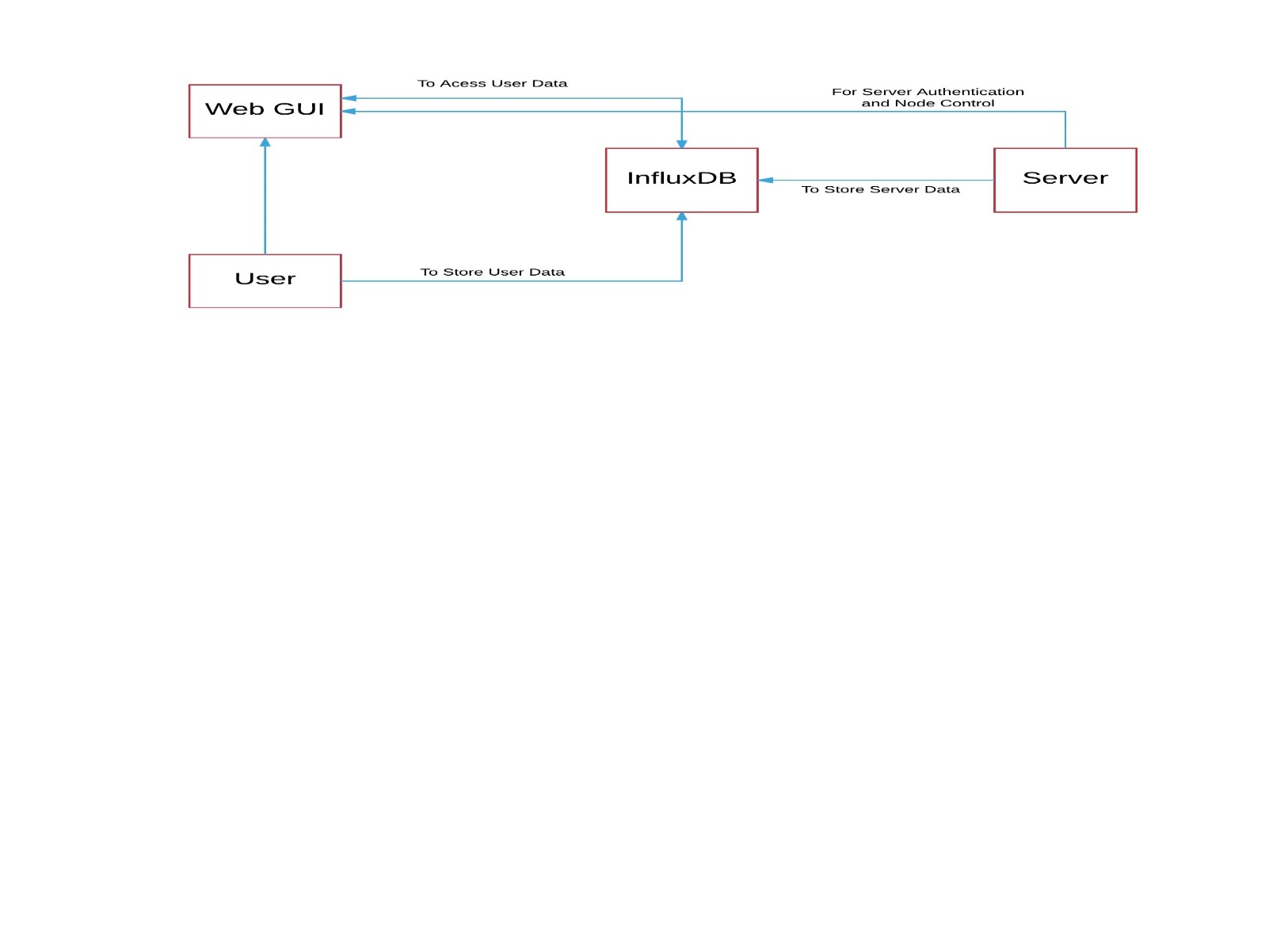
**3.2) DATABASE:**

Fig: Database

This module represents the database management tools used in this product. The MySQL database contains user authentication data, status and uptime information inserted into their respective service tables. Different tables are created in InfluxDB for different nodes with different datatypes. For example: load, CPU utilization, Bandwidth.

**3.3) BACKEND:**

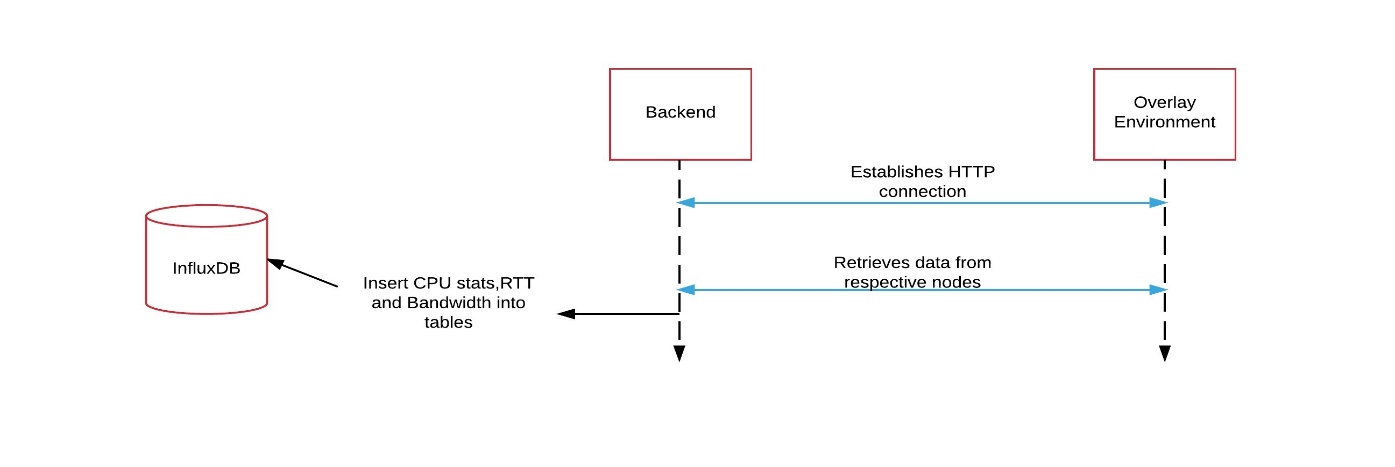


Fig: Backend

This module represents the backend of the system architecture. The backend script remotely connects to the Overlay network monitoring system using HTTPS connection. This backend Perl script uses variables that contain host IP address, private key and passphrase to establish a secure connection

It retrieves status and inserts into the MySQL database into tables created with different names for each service. It also facilitates restart of service upon request by user from frontend. The restart is required for debugging with encryption disabled.

**4) REQUIREMENTS:**

The user must be able to select the service to monitor. The details of services to be monitored will be provided to the user, demanded in accordance with certain fixed regulations. The project requirement defines both user requirements and system requirements. These are discussed briefly in the list of requirements section.

**4.1. User requirements:**

This section describes the services provided for the user by the product. This section is further divided into functional and non-functional requirements subsections.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| S. No | User Requirements | Type | Module | Dependencies | Identification | Description |
| 1 | E-mail Confirmation | Functional | Back End | Req\_SYSF1  Req\_SYSF2  Req\_NFSYS2 | Rq\_USRF1 | To confirm user registration. |
| 2 | Syslog Facility | Functional | Back End | Req\_SYSF1  Req\_NFSYS2 | Rq\_USRF2 | To track multiple node activities. |
| 3 | Web GUI | Functional | Front End | Req\_SYSF1  Req\_SYSF2 Req\_NFSYS2 | Rq\_USRF3 | For user interface. |
| 4 | Add/Remove Nodes | Functional | Front End | Req\_SYSF1 Req\_SYSF2  Req\_NFSYS2 | Rq\_USRF4 | Management of new user account and registered nodes. |
| 5 | Login Authentication | Functional | Front End | Req\_SYSF1 Req\_SYSF2 | Rq\_USRF5 | To provide secure access for authorised users. |
| 6 | RESTful API with Flask | Functional | Front End | Req\_SYSF1 | Rq\_USRF6 | To handle requests. |
| 7 | Grafana | Functional | Front End | Req\_SYSF2 | Rq\_USRF7 | Displays statistical graphs. |
| 8 | Restart (Encryption Disabled) | Functional | Back End | Req\_SYSF1 Req\_NFSYS2 | Rq\_USRF8 | Restating the system for debugging purpose. |
| 9 | Documentation | Non-functional |  |  | Rq\_NFUSR2 | Includes user manuals and reading instructions. |

**4.2. System requirements**:

This section provides details about the system requirements. These are technical requirements that complement the user requirements and provide information for design and implementation of product. In the same manner as in user requirements, this section is also divided into subsections of functional and non-functional requirements.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S. No | System Requirements | Types | Module | Identification | Description |
| 1 | Web Server | Functional | Front End | Req\_SYSF1 | To interface between the user and the web. |
| 2 | Databases (Login credentials and Statistics) | Functional |  | Req\_SYSF2 | To store login credentials and user data. |
| 3 | Operating Systems | Non-functional |  | Req\_NFSYS1 | Ubuntu 16.04 is used. |
| 4 | Scripting Languages | Non-functional |  | Req\_NFSYS2 | They are used to build the tool. |
| 5 | Overlay Network | Functional | Back End | Req\_SYSF3 | It monitors the nodes present a mesh network. |

**5) REFERENCES**

1. [https://opennetworkingusergroup.com/wp](https://opennetworkingusergroup.com/wp-content/uploads/2015/05/ONUG-Overlays-Whitepaper_Final1.pdf)- [content/uploads/2015/05/ONUG-Overlays-Whitepaper\_Final1.pd](https://opennetworkingusergroup.com/wp-content/uploads/2015/05/ONUG-Overlays-Whitepaper_Final1.pdf)f

2. [http://www.mi.parisdescartes.fr/~jelias/papers/PoliMi\_Thesis\_JocelyneEl](http://www.mi.parisdescartes.fr/~jelias/papers/PoliMi_Thesis_JocelyneElias.pdf)i  [as.pd](http://www.mi.parisdescartes.fr/~jelias/papers/PoliMi_Thesis_JocelyneElias.pdf)f