

May1618

Graphalyzer

workiva

A Graph Visualization and Analysis Tool

Andrew Bowler - Webmaster
Alberto Gomez - Comms Lead

Michael Sgroi - Key Concept Holder
Taylor Welter - Team Leader
Richard White - Key Concept Holder

Dr. Simanta Mitra - Advisor
Ross Hendrickson - Client

Problem

Companies are constantly dealing with Big Data. Big Data is very difficult to understand or interpret, and analysis of the data is far from trivial.

Sometimes data is often structured such that pieces of the data have relationships with one another, further increasing the complexity.

How can we analyze all of the connections buried within this data?

Solution

Graphalyzer provides users with an interactive visual graph of the data through their web browser. The user can explore the graph and perform analysis on the data through searching, filtering, and drawing of subgraphs.

Graphs make this type of data easier to understand because they highlight the structural behavior of the data, and make it easy to find members of high importance, such as interconnectivity or having specific properties.

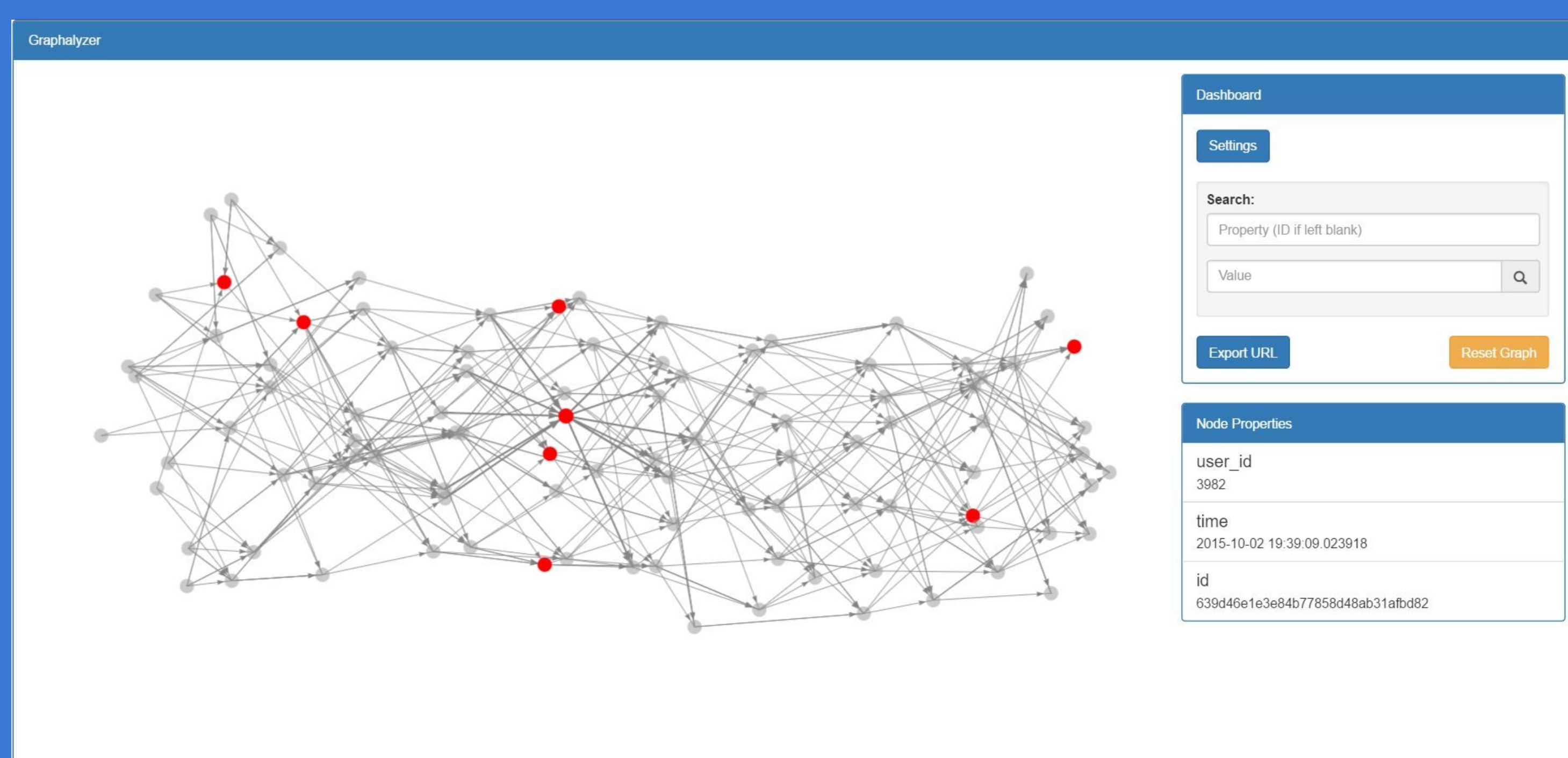


Figure 1. Filtering Nodes

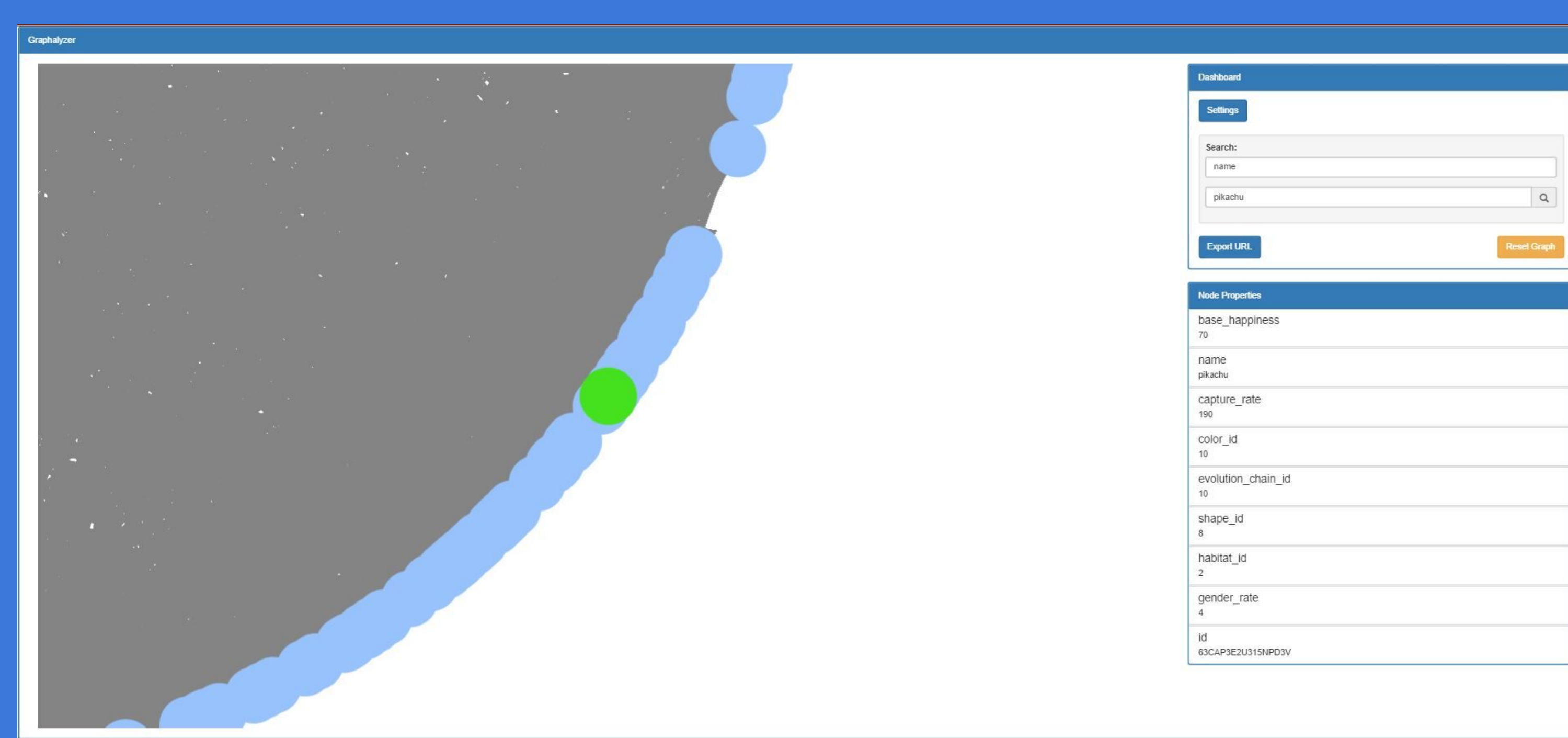


Figure 2. Searching for a Node

Requirements

- Visualize an interactive graph in a web browser
- Display graphical analysis based on user settings
- Handle graph data uploads via REST
- Handle data up to ~40GB in size

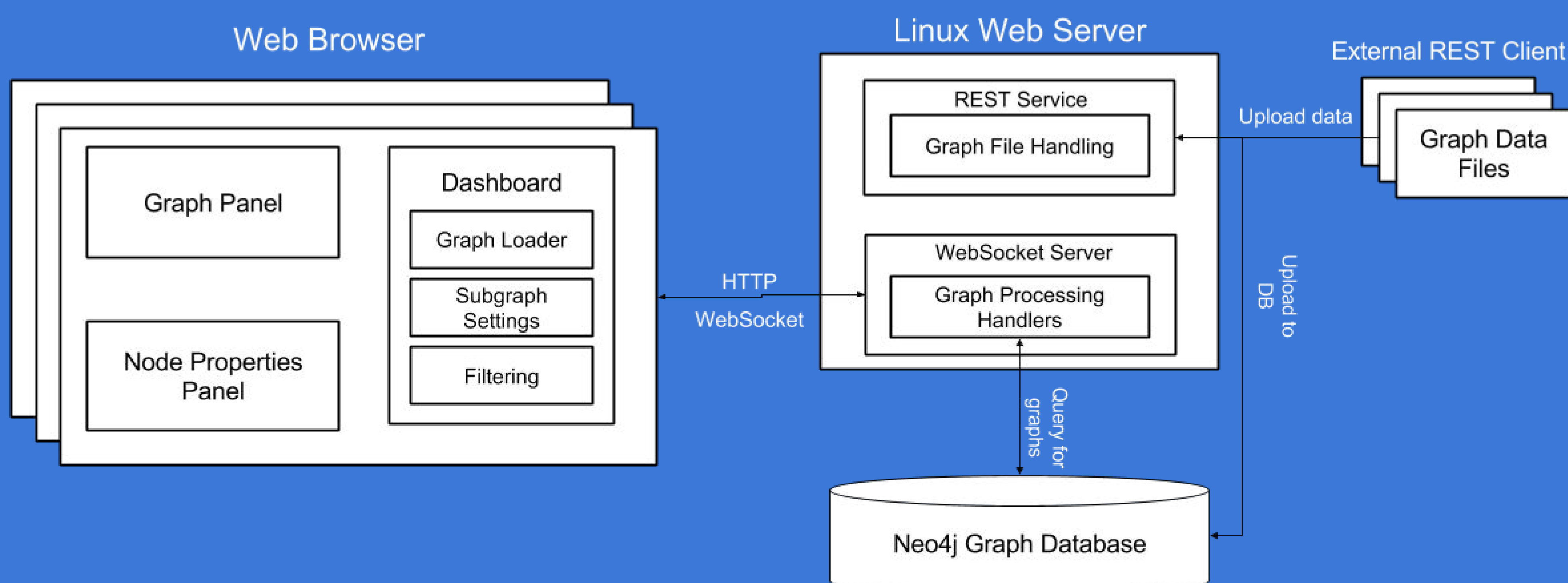
Technical Details

- React.js - frontend framework
- Vis.js - graph library
- Python - backend language
- Neo4j - graph database
- Linux Web Server - hosting application

Challenges

- Change of tech stack during the semester
- Processing gigabytes of data while minimizing impact to interactivity
- Implementing a test plan with the provided tools

Project Design



- Web Browser
 - Provides user interface for graph visualization and analysis
 - Displays interactive graph
 - Receives data from the server
- Linux Web Server
 - Graph processing and uploading written in python
- Neo4j
 - Database specialized for graph operations and data structures
- Jest to write unit tests for the code.
- PyUnit to write separate tests for the server.