

# Real-Time Route Optimization

## Summary

- **Problem:** Fusing real-time information (location; route; weather;...) for the purpose of improved management of fleets during snow operations.

- **Approach:** Develop a system/tools for: (1) Integrating heterogeneous data; (2) Provide visual display of parameters of interest; (3) Incorporate the decision-making.

## Requirements

### Functional:

- Store history data
- Gives route suggestions
- Shows truck's path in the map
- Authentication upon type of user
- Simulate and generate truck's data
- Shows trucks' information upon request
- Shows weather conditions near truck location
- Shows history path and data upon request

### Non-Functional:

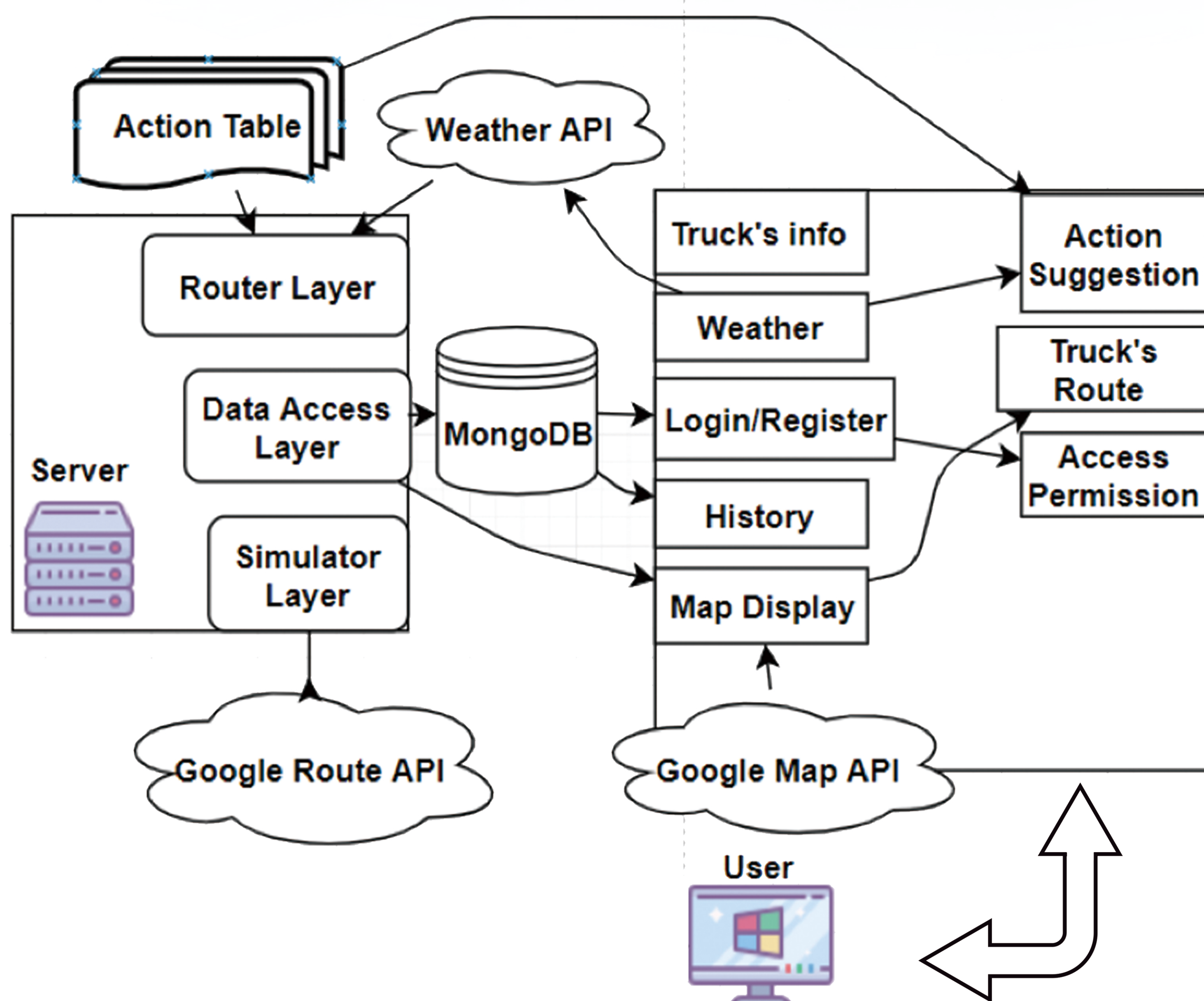
- Response time from server under 200ms
- Handling data without delay when data increases
- Contents and displays should be easy to understand

## Testing and Evaluation

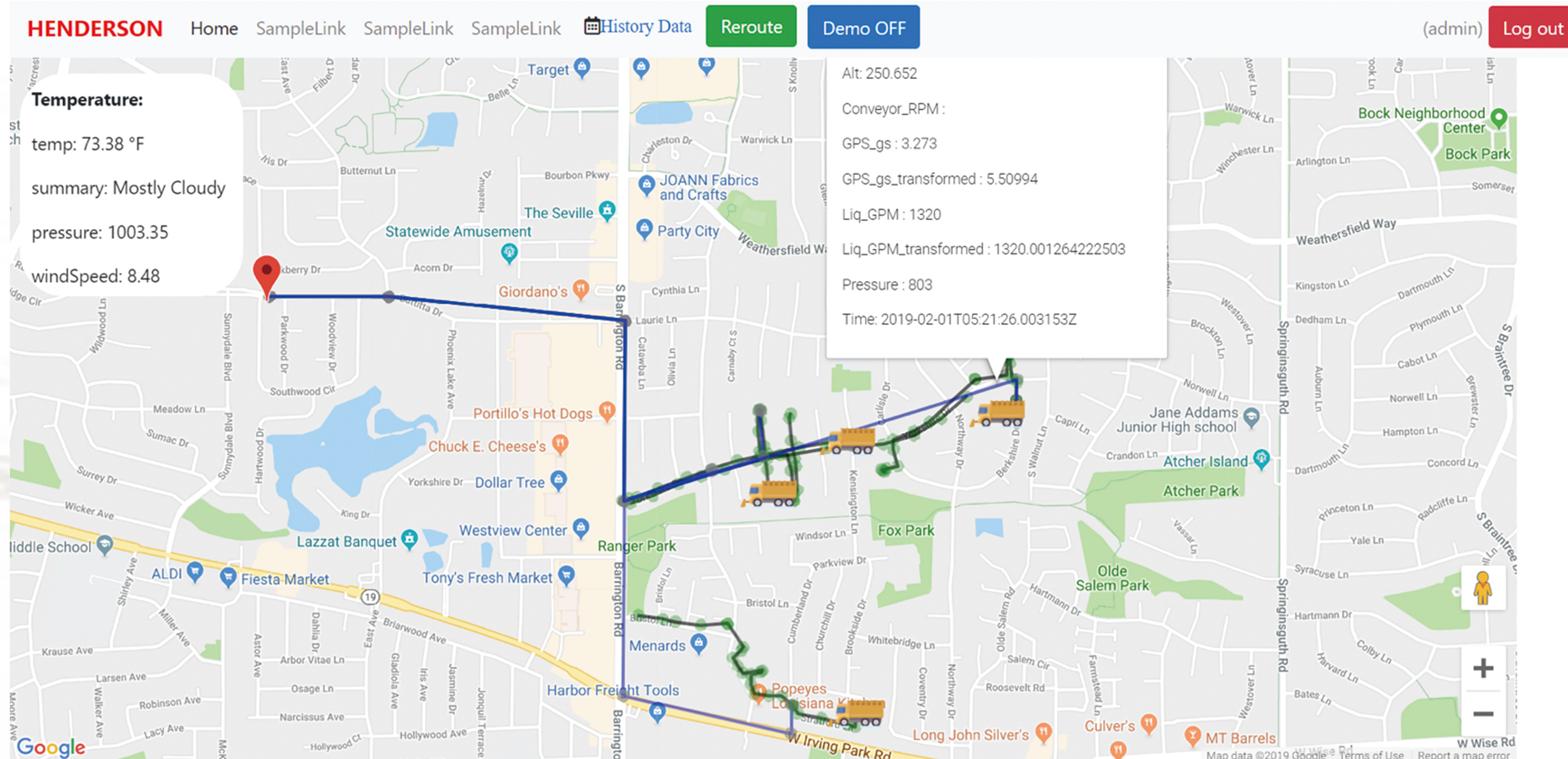
- Stress test: send request by JMeter to AWS server
- Performance test: use JQuery to test the load of front-end page
- User test: real sensor was deployed on our car
- System test: implement test code based on black box test policy
- Integration test: for different platform

## System Architecture and Design

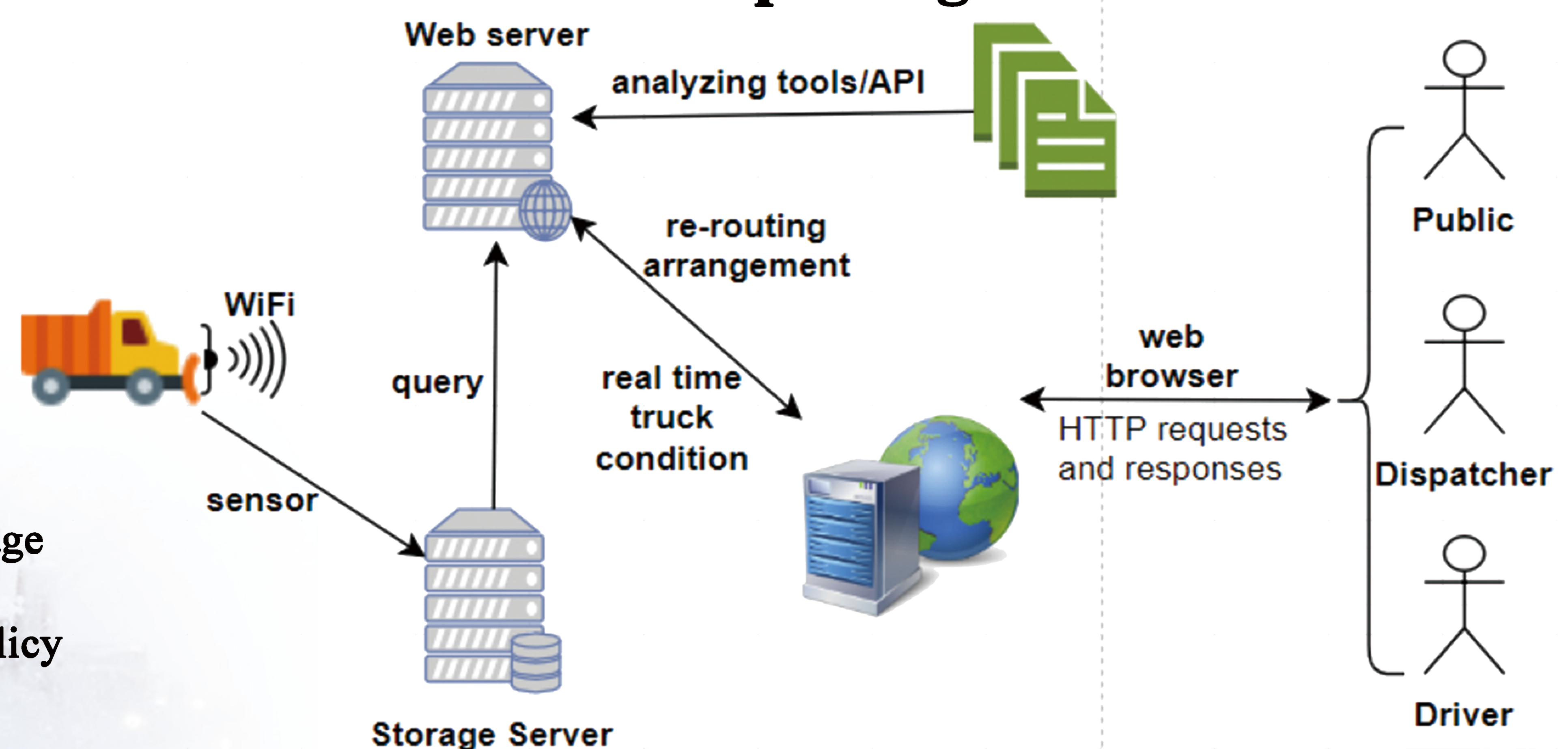
- Re-routing algorithm depends on Action Table, Google Route API, and weather API
- Three different layers of access and map display: driver, dispatcher, public
- Database stores suggested route and history data



## Main Page



## Concept Diagram



## System Implementation

- Web application framework: AngularJS
- AWS server: NodeJS
- Google map API
- Google Route API
- Dark Sky Weather API
- Open Weather API
- Database: Mongo DB, JSON format data

## Constraints

- Unable to get some physical data (road conditions from radar, street camera) – due to government regulations.
- Upload speed: so far 4 trucks can be used in our case.

## Standards

- Protocols: agile (IEEE, ABET)
- Version Control System: Git, Google software suite
- Code Review: peer review (feature branches)
- User test: real sensors deployed

## Budget

- Dark sky weather API: \$0.0001 per call
- AWS (Amazon Web Server): \$650 per year

## Conclusion

We have accomplished

- Receive, store, display information of truck
- Simulator: generate data and transit from server to front end
- 3 different layers of access
- Routing suggestion and route display based on Action Table and real time weather
- In the future, a more comprehensive algorithm could be considered for re-routing design

## Team members :

Junjie Wen Zhanghao Wen  
Yuhang Xie Xinhe Yang Tianhao Zhao

## Clients :

Adam Ryan  
Henderson Products Inc.

## Faculty Advisors :

Dr. Goce Trajcevski