MICHIGAN STATE UNIVERSITY

Project Plan

Kubernetes Cluster Inspection Tool

The Capstone Experience

Team Google

Dave Ackley
Linghao Ji
Guillermo Jimenez
Haylee Quarles
Casey Schneider
Ben Whitelaw

Department of Computer Science and Engineering
Michigan State University

Spring 2019

From Students…
…to Professionals

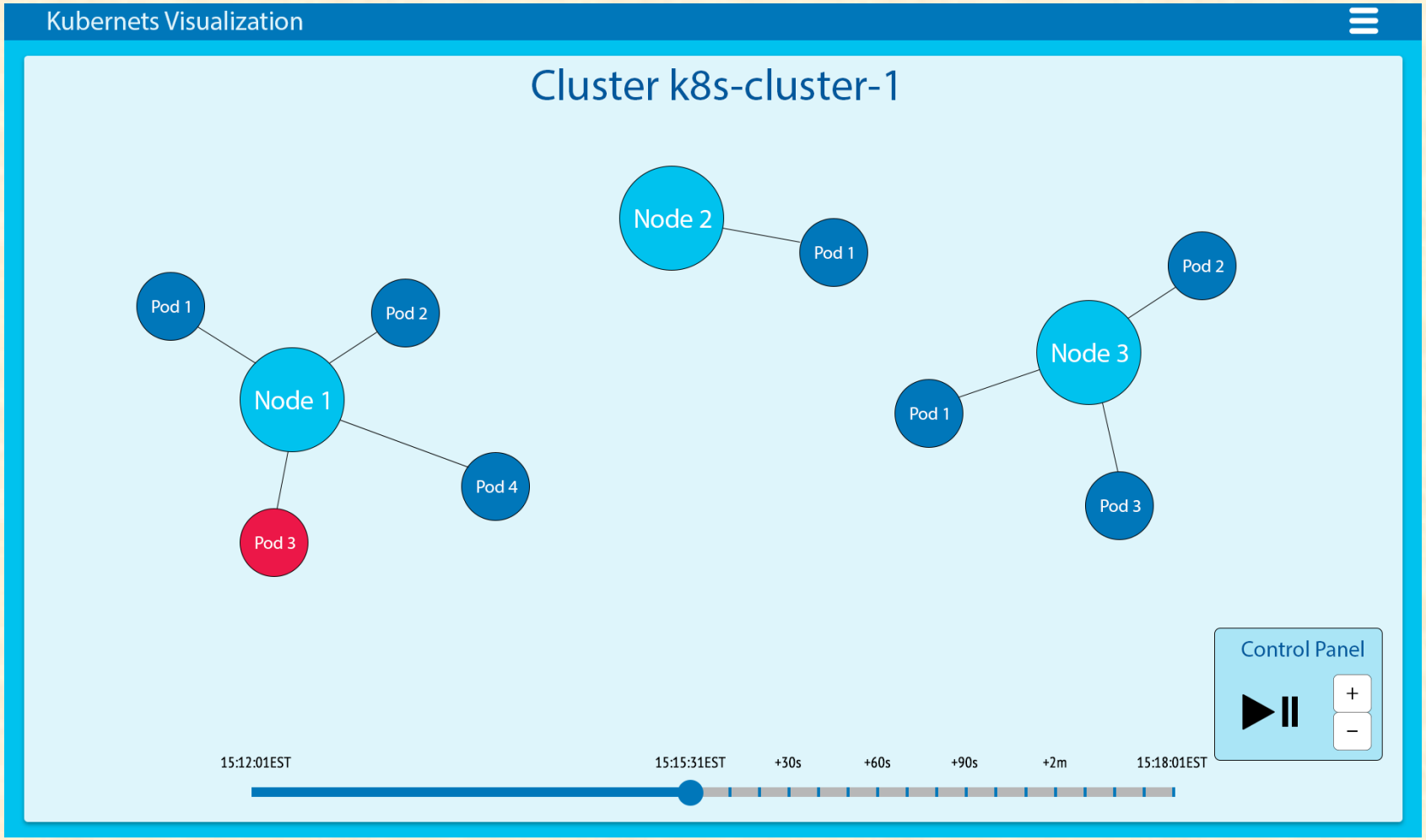
Functional Specifications

- Provide Robust Inspection Tool of Kubernetes Cluster
  - Both Current and Historical State
- Gather All Cluster Data into Single Interactive View
  - Provide Health, Performance, and Resource Changes
  - Ingest with Kubernetes Stackdriver Agent
  - Trace Lifecycle of Nodes, Pods and Containers
- Implement Easily on Any GCP Kubernetes Cluster

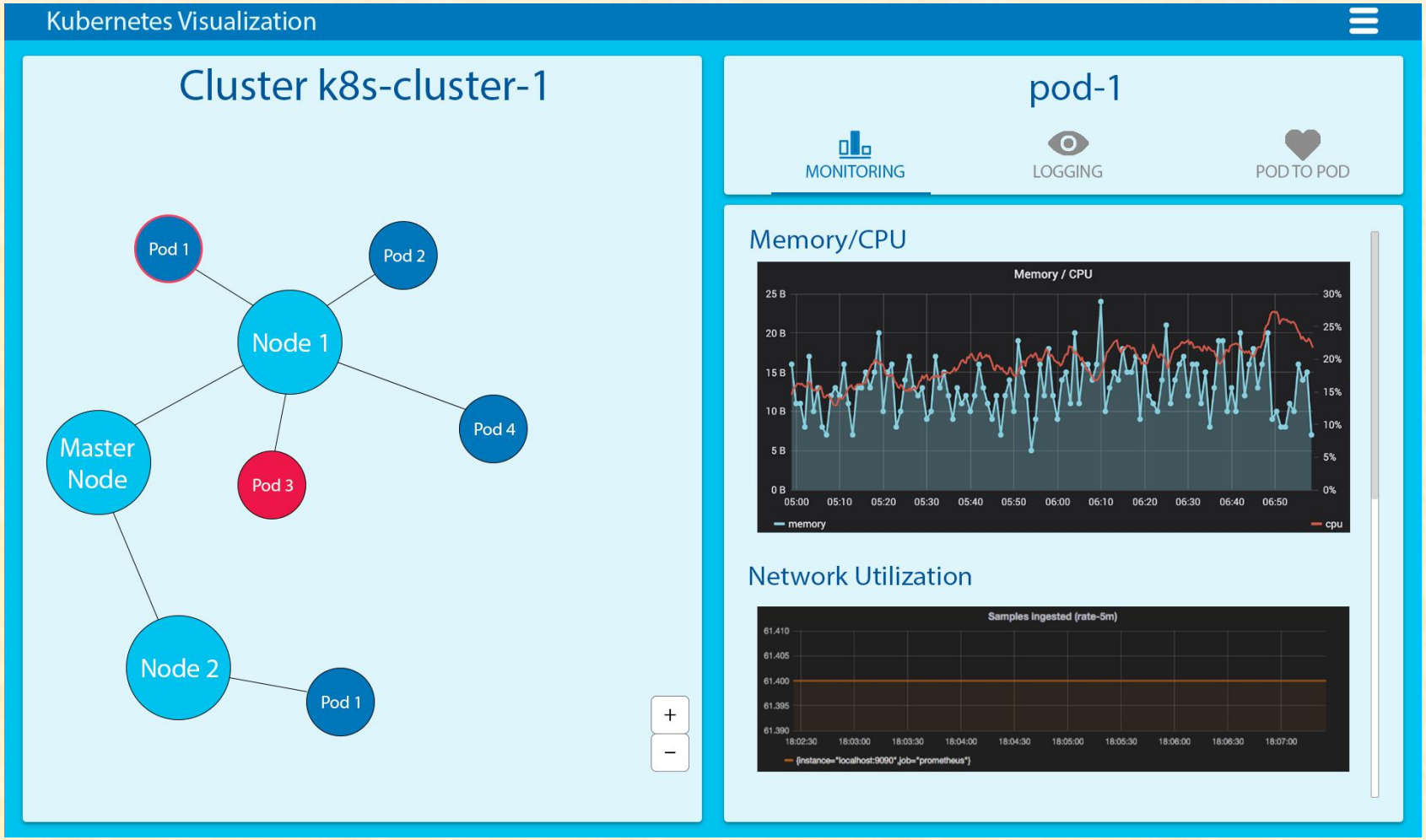
Design Specifications

- Reactive Single Page Application
- Cluster Visualization
  - Monitoring and Logging
    - Grafana Dashboard
  - Pod-to-Pod and Service-to-Service Communication
- Historical View Playback
- Focus: Assist Cluster Troubleshooting
- Material Design Framework

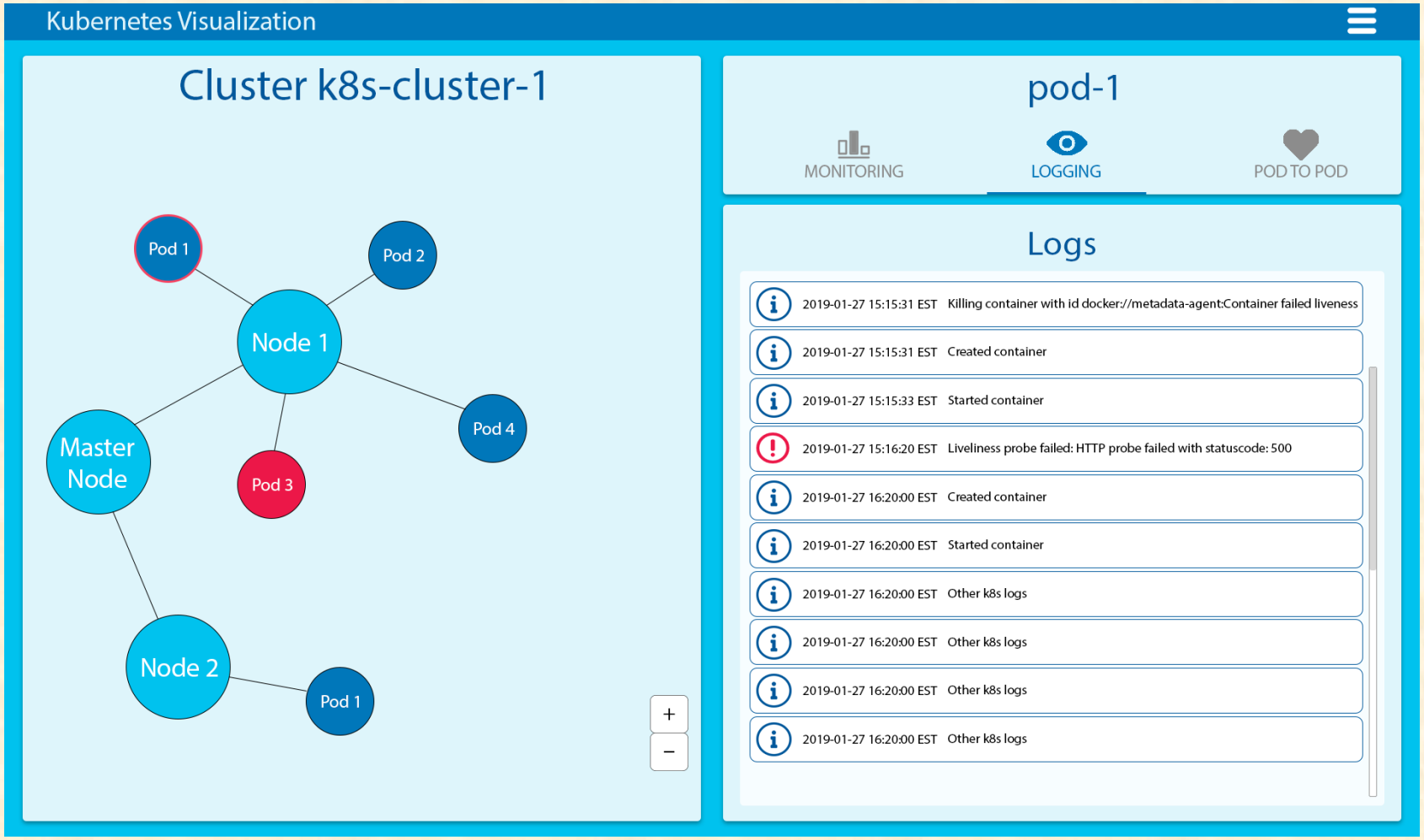
Screen Mockup: Interactive Cluster Map



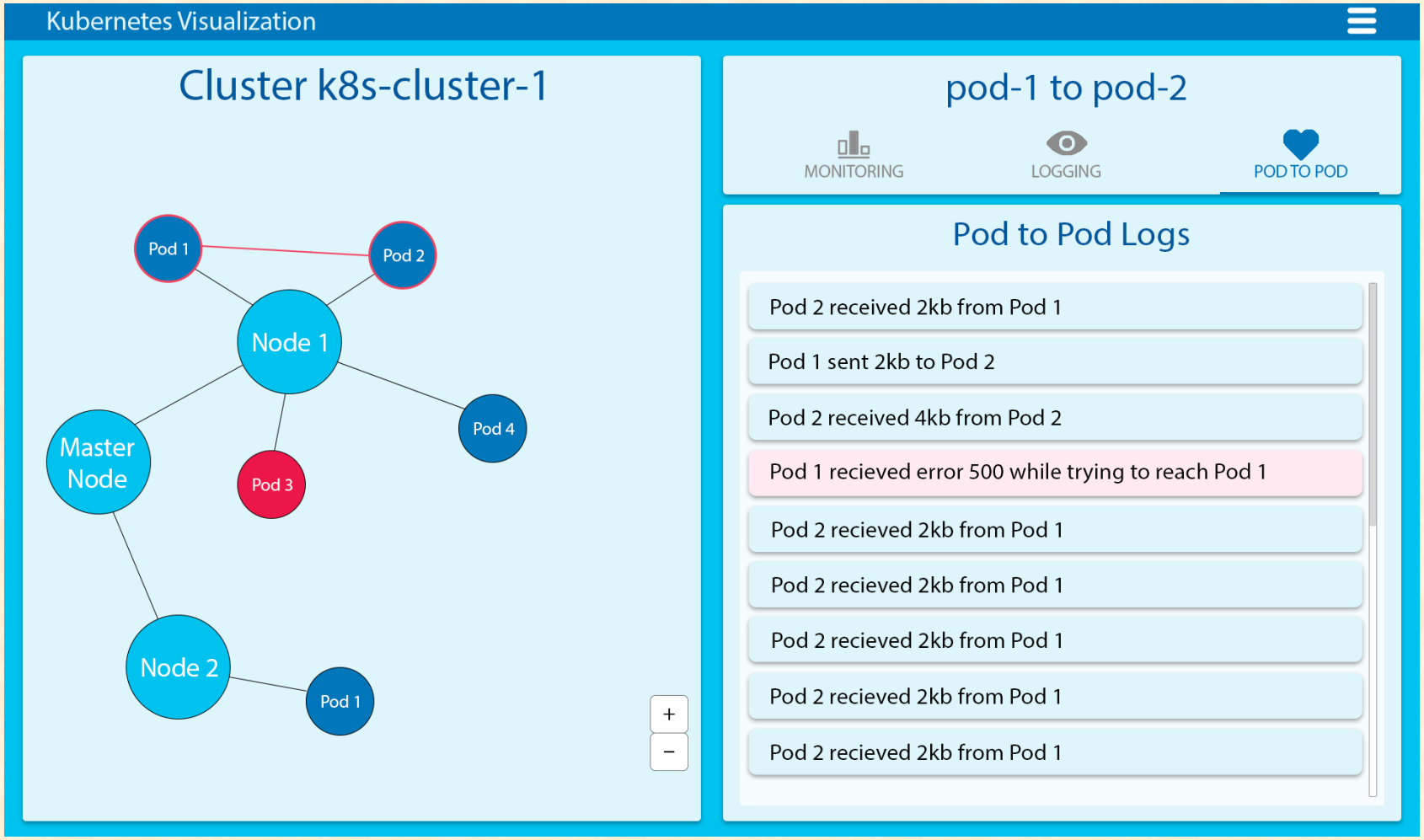
Screen Mockup: Drilldown Monitoring View



Screen Mockup: Drilldown Logging View



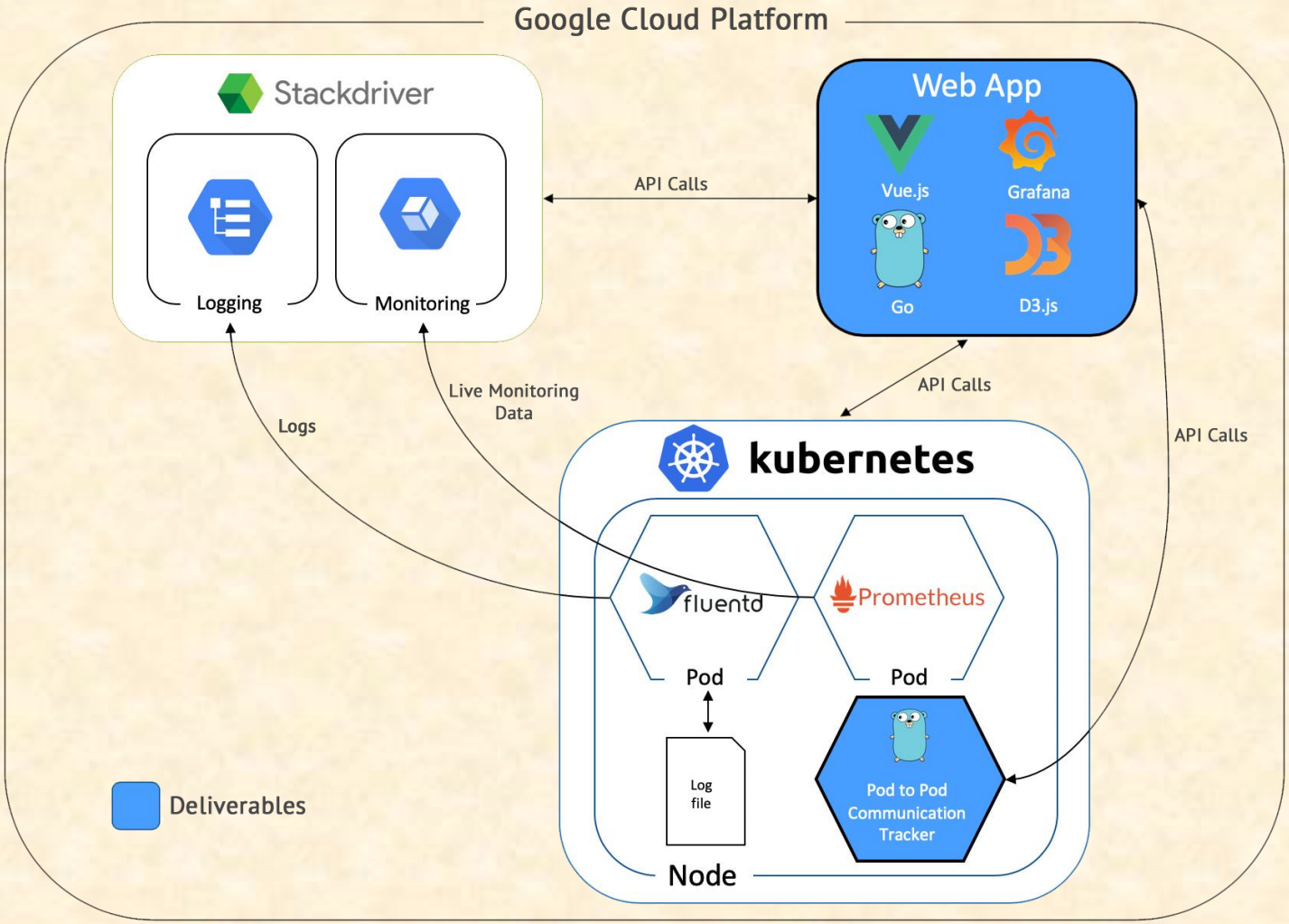
Screen Mockup: Pod to Pod View



Technical Specifications

- Web Application
  - Metrics and Logging Visualization Dashboard
  - Interactive Cluster Map and Events Timeline
- Pod-to-Pod Communication Tracker
  - Daemon Set on each node in the cluster
  - RESTful API for the web app to access the data

System Architecture



System Components

- Frontend
  - Vue.js
  - Grafana
  - D3.js
- Backend
  - Go
- Software Platforms / Technologies
  - Kubernetes
  - Stackdriver APIs
    - Monitoring
    - Logging
    - Fluentd and Prometheus

Risks

- Feasibility of Pod-to-Pod Communication Monitoring
  - Description: There is no other product on the market that is able to achieve this feature – not sure if it is even possible
  - Mitigation: Adding a Daemon Set to each node in the cluster
- Cost Effective Data Caching Algorithm
  - Description: Making calls to the Stackdriver API racks up a charge
  - Mitigation: Data obtained from the APIs will be stored in a cache for easy retrieval
- Unfamiliarity with Needs of the End User
  - Description: No one on the team uses Kubernetes in our daily lives, can't easily anticipate what features would be most useful
  - Mitigation: Reach out to Kubernetes users for feedback and input
- Playback Button to See Historical Cluster Events
  - Description: Not sure if this feature is possible with available technology
  - Mitigation: Use other applications with similar features as a model, and do some testing of this feature on a small scale using javascript

Questions?