Team 3: Sam Bostian, Michael Rizig, Charlie Mclarty, Brian Pruitt, Allen Roman

GCPS REAL-TIME BUS MONITORING

SAM BOSTIAN

- **Role:** Team Leader, RHEL9 Install
- **Major:** Computer Science, B.S.
- **Graduation:** December 2024
- **Experience:**
  - Research Project on the benefits of Parallel Processing

sbostian@students.kennesaw.edu

MICHAEL RIZIG

- **Role:** Primary Developer
  - Kafka, Backend, MSSQL
- **Major:** BS&MS Computer Science
- **Graduation: Spring 2025**
- **Experience:**
  - IT Intern at DCCU

mrizig@students.kennesaw.edu

CHARLIE MCLARTY

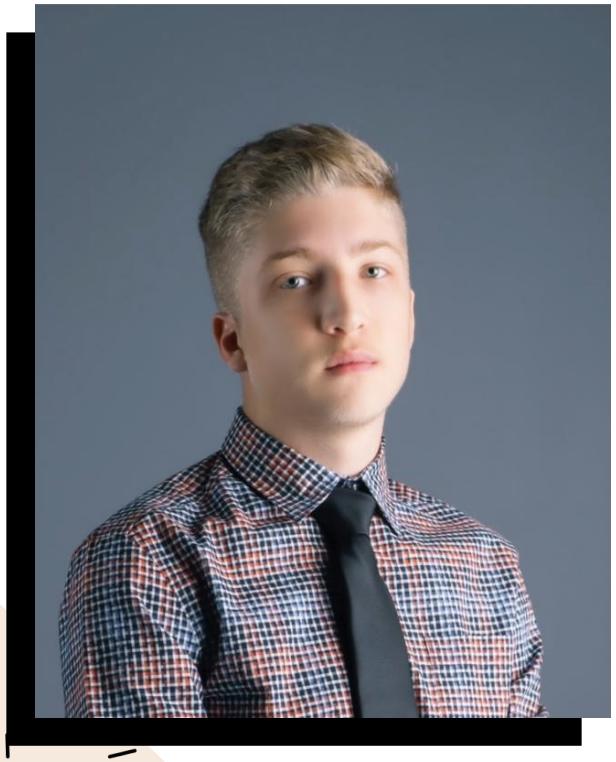
- **Role:** Developer
  - Data Generation & Simulation, QA Testing, Confluent Dashboard
- **Major:** Computer Science
- **Graduation:** December 2024
- **Experience:**
  - Research on edge computing energy efficiency

cmclart4@students.kennesaw.edu

BRIAN PRUITT

- **Role:** Documentation
- **Major:** Computer Science, B.S.
- **Graduation:** Fall 2024
- **Experience:**
  - Interned as a Software Engineer for The Home Depot

bpruitt9@students.kennesaw.edu



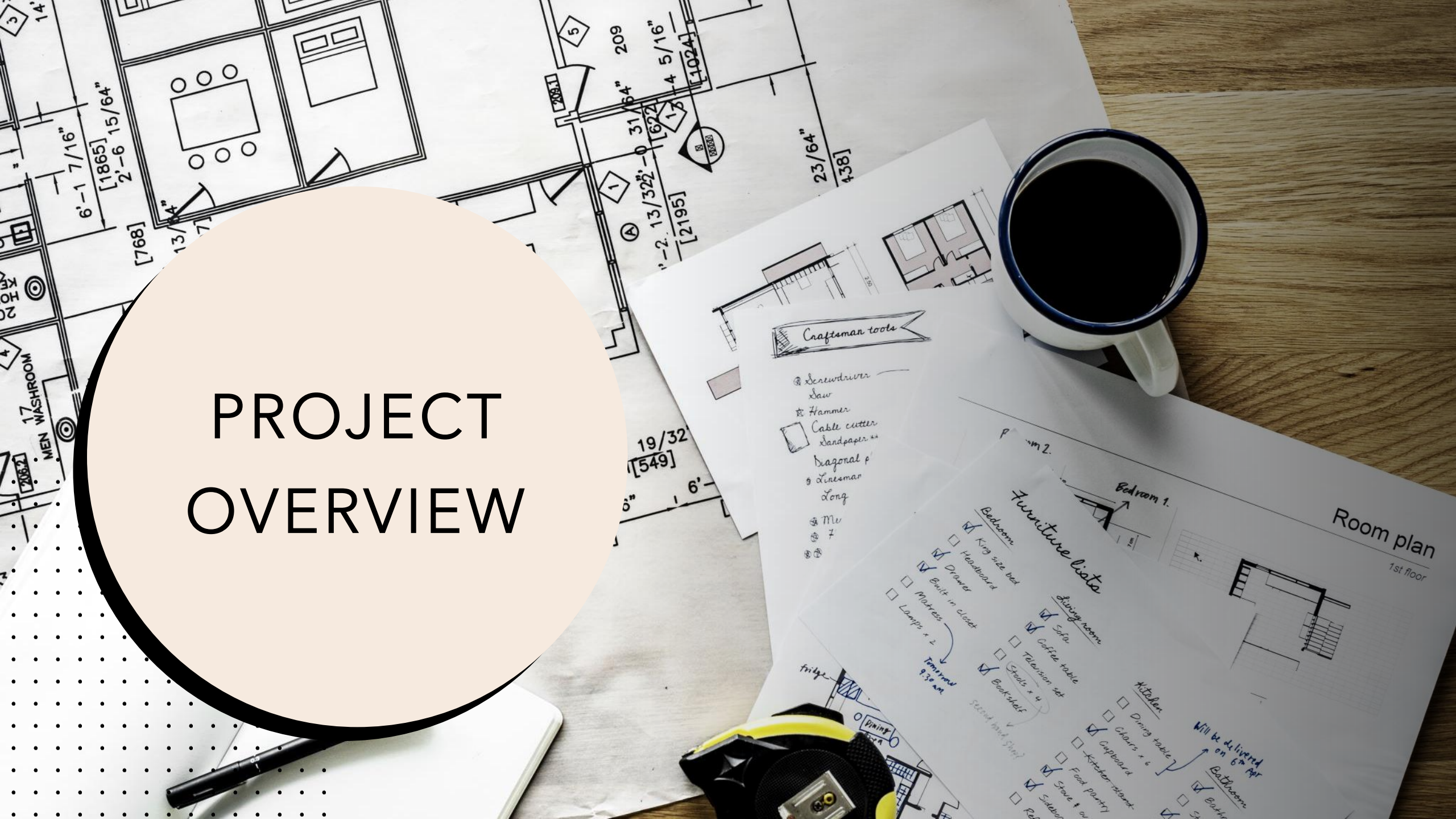
ALLEN ROMAN

- **Role:** Developer
  - Containerization, Deployment, Confluent Dashboard
- **Major:** Computer Science, B.S.
- **Graduation:** May 2025
- **Experience:**
  - Software Developer Intern at ADP

aroman14@students.kennesaw.edu

MEETING AGENDA

- Project Overview
- Containerization & Deployment
- Demonstration
- Testing & Documentation


PROJECT
OVERVIEW
Craftsman tools
Screwdriver
Saw
Hammer
Cable cutter
Sandpaper
Diagonal
Linesman
Long
Me
7
Furniture lists
Bedroom
King size bed
Headboard
Drawer
Built in closet
Mattress
Lamps x 2
Tomorrow 9:30 AM
Bedroom 1.
Living room
Sofa
Coffee table
Television set
Bookshelf
Stools x 4
Second hand store
Kitchen
Dining table
Chairs x 6
Cupboard
Kitchen utensil
Food pantry
Stove
Will be delivered on 6th Apr
Bathroom
Room plan
1st floor
fridge
Dining area
MEN WASHROOM

PROBLEM STATEMENT

Currently GCPS uses Samsara REST API calls to poll for bus data approximately every 5 seconds.

Due to the vast bus fleet and how many calls made per minute GCPS makes, they are currently in the top 1% of all of Samsara's API calls.

GCPS would like to switch to using the Samsara Connector and store this bus data into an SQL relational database that can be called by other applications for near real-time data processing.

PROJECT OBJECTIVES

- Transition from Samsara API polling to Kafka-based event streaming for real-time bus tracking, creating an in-house solution for efficiency and data security.
- Efficiently process and validate bus telemetry data and store it in SQL Server.

TECH STACK

To ensure an easy redevelopment process, we restricted ourselves to a technical stack of software used by GCPS:

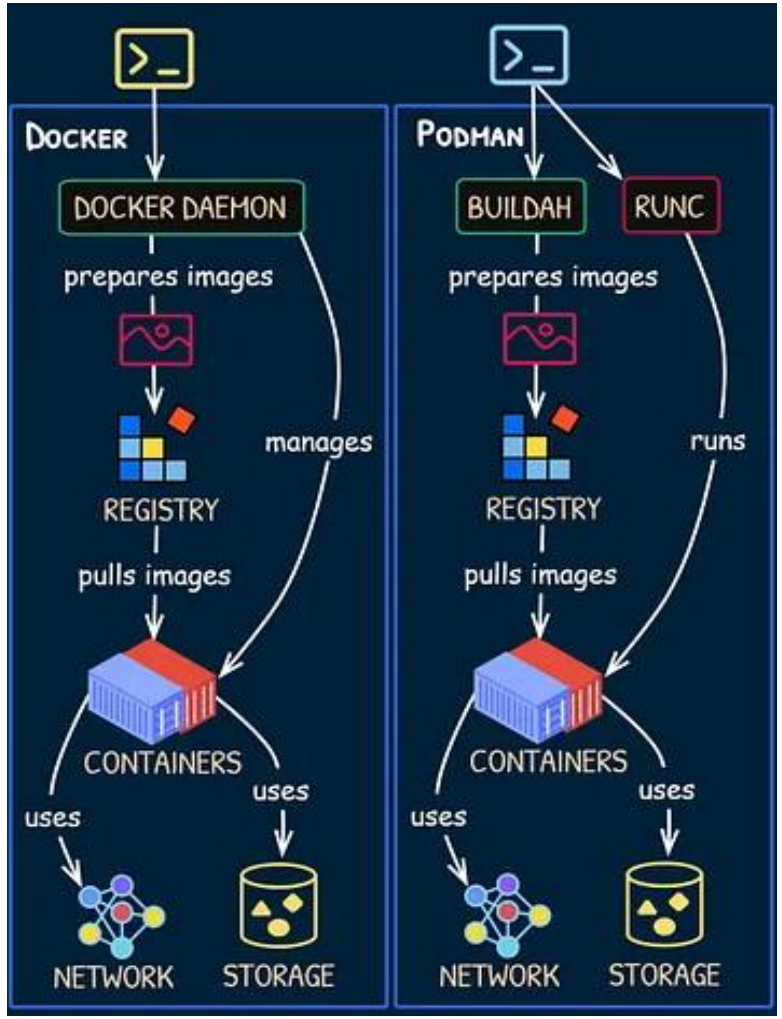
CONTAINERIZATION & DEPLOYMENT

PODMAN

We chose Podman for containerization due to its security benefits and compatibility with Red Hat environments.

Application runs a Pod with a container for the producer-side and one for the consumer-side

Podman’s rootless containers provide an extra layer of security, which is ideal for production use.



POD CONFIGURATION

The *pod.yml* file defines the pod structure, networking, and resource allocations, enabling smooth interaction between containers:

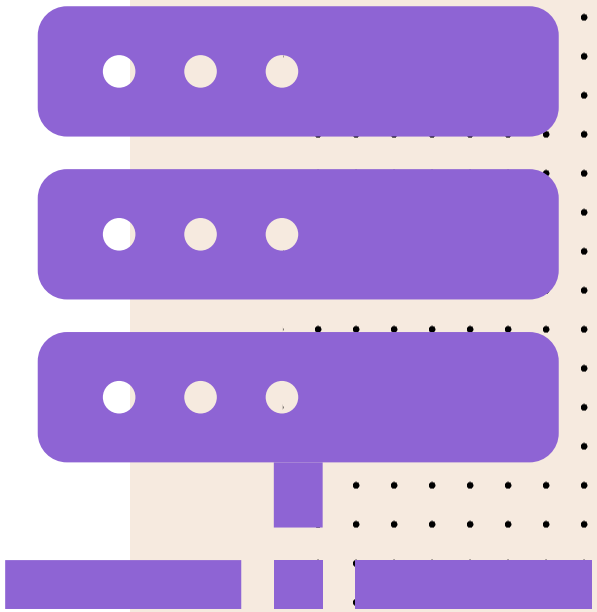
Producer Container (bus-project-producer)

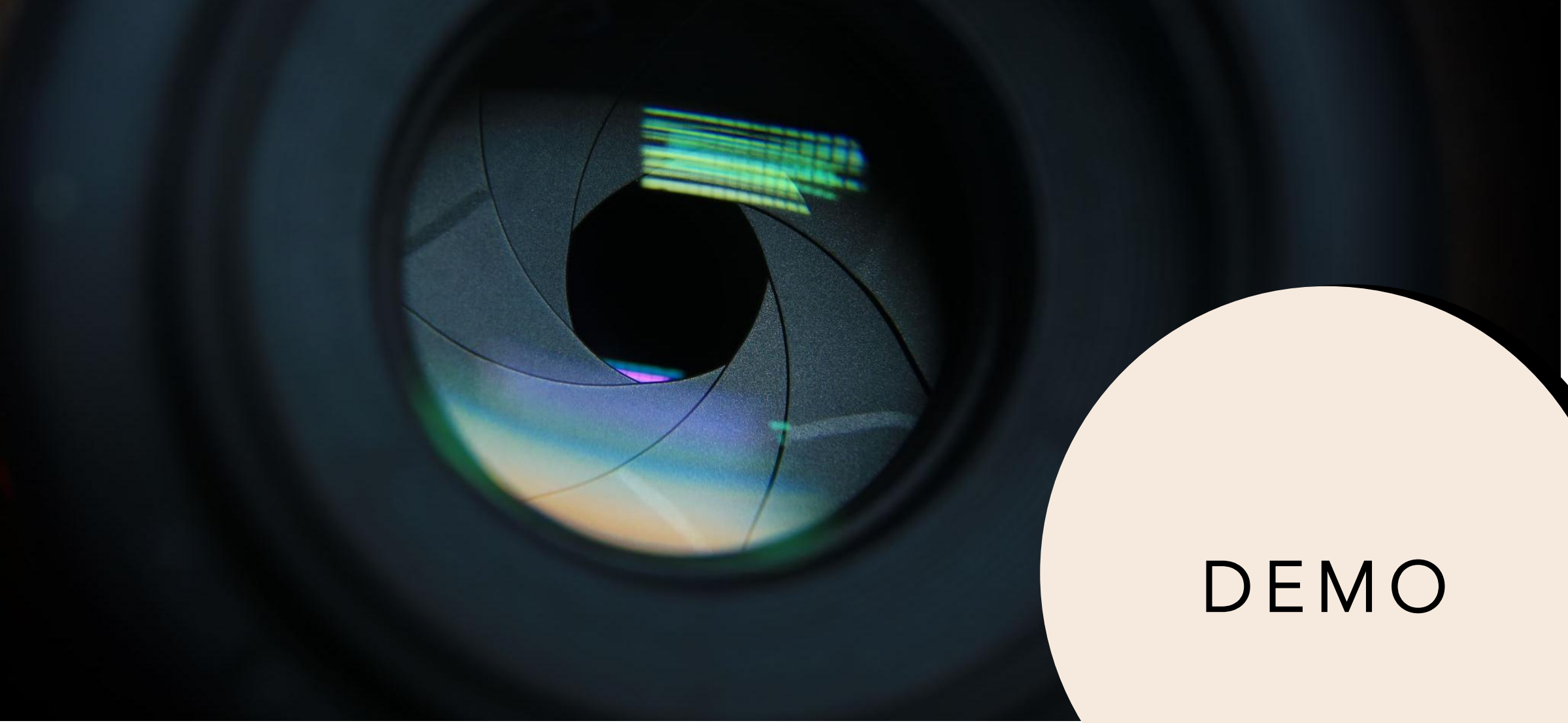
- Handles Kafka event streaming, sending bus data to the topic.

Web Container (bus-project-web)

- Hosts the web interface for user interaction and data visualization.

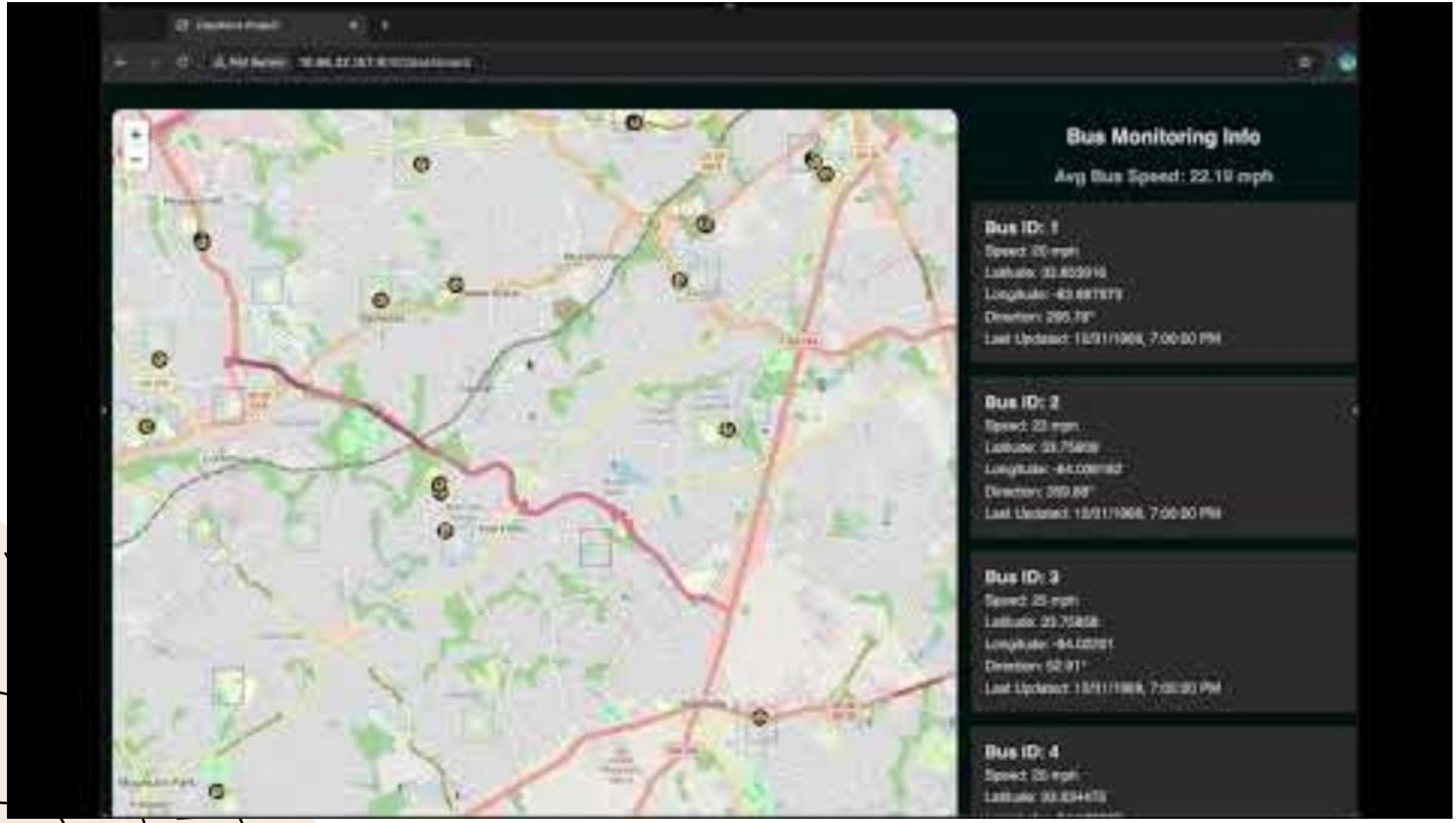
Together, these containers operate within a Podman pod, ensuring efficient data flow and container management.





DEMO

https://www.youtube.com/watch?v=UMjljWKBEt0


Bus Monitoring Info
Avg Bus Speed: 22.19 mph
Bus ID: 1
Bus ID: 2
Bus ID: 3
Bus ID: 4

TESTING & DOCUMENTATION

SYNTHETIC DATA GENERATION

Gwinnett county represented as a strongly connected graph

GWINNETT COUNTY MAP CONVERTED TO DIRECTED GRAPH

DYNAMIC ROUTE GENERATION FROM EACH SCHOOL

DATA COLLECTOR RECEIVES UPDATES FROM EACH BUS AND STREAMS TO THE KAFKA PRODUCER

TIME SIMULATION TO DEMONSTRATE MORNING BUS ROUTES

SOFTWARE TEST REPORT

Requirement	Description	Pass	Fail	Severity
Data Validation	Filter data and flag anomalous / 'stale' data	✓		**Low**
Real-time data ingestion through Kafka	Verify that the Kafka producer sends data in real time	✓		**High**
Container initialization	Check if all containers start without errors	✓		**High**
Inter-container communication	Confirm Kafka-to-SQL data flow in pod environment	✓		**High**
Pod shutdown and cleanup	Ensure stop_pod.sh script removes all containers/pods	✓		**Medium**
Monitoring setup (optional feature)	Verify monitoring tool displays container status	✓		**Low**
Scalability test with 2,000 bus data points	Check system performance under high data volume	✓		**High**
Consistent container deployment	Test pod deployment on different environments	✓		**Medium**

DOCUMENTATION

README.md Overview

The README provides setup instructions, detailing installation, configuration, and initial deployment.

Configuration and Usage

Instructions for modifying environment variables, scaling the system, and handling API requests are included to support GCPS's needs.

Optional Monitoring

Optional monitoring configuration is provided in the documentation, allowing GCPS to track container performance and ensure system stability.

QUESTIONS