Development Team

Arjun Vinod
- Web App
- PCB Design

Brian Li
- PCB Design
- Enclosure Design

Nicholas Tran
- Firmware
- Backend

Bryan Olivares
- Frontend
- Backend Infra

Hyun Kyum Kim
- Frontend
- Backend Infra
Problem Statement
Lost in Translation
Lost in Translation

Cars already talk to us with colorful and creative icons on the dash
Lost in Translation

Cars already talk to us with colorful and creative icons on the dash.

But for a team, this **single stream** of information is a **bottleneck**.
Lost in Translation

Cars already talk to us with colorful and creative icons on the dash.

But for a team, this single stream of information is a bottleneck.
Lost in Translation

Cars already talk to us with colorful and creative icons on the dash.

But for a team, this **single stream** of information is a **bottleneck**.

The operator of this vehicle has the additional responsibility for **gathering**, **organizing**, and **storing** this data.
Lost in Translation

Cars already talk to us with colorful and creative icons on the dash.

But for a team, this **single stream** of information is a **bottleneck**.

The operator of this vehicle has the additional responsibility for **gathering**, **organizing**, and **storing** this data.

Most importantly, to develop **insights** using this data.
Proposed Solution
End-to-End Vehicle Data Tracking
End-to-End Vehicle Data Tracking

Collect live vehicle data

- [12.345°N, 21.345°W, 45 mph, 70°F, ...]
- [37.375°N, 19.345°W, 47 mph, 67°F, ...]
- [47.342°N, 17.345°W, 55 mph, 57°F, ...]
End-to-End Vehicle Data Tracking

Collect live vehicle data

Upload it to the cloud

[12.345°N, 21.345°W, 45 mph, 70°F, ...]

[37.375°N, 19.345°W, 47 mph, 67°F, ...]

[47.342°N, 17.345°W, 55 mph, 57°F, ...]
End-to-End Vehicle Data Tracking

Collect live vehicle data
Upload it to the cloud
Serve to the user

[12.345°N, 21.345°W, 45 mph, 70°F, ...]
[37.375°N, 19.345°W, 47 mph, 67°F, ...]
[47.342°N, 17.345°W, 55 mph, 57°F, ...]
Tracker Module Installation
Web App Overview
Implementation Details
GPS data is embedded in a PVT (Position/Velocity/Time) frame fetched periodically from the microcontroller’s built-in GPS/LTE modem.
Location Data

GPS data is embedded in a PVT (Position/Velocity/Time) frame fetched periodically from the microcontroller’s built-in GPS/LTE modem.
Location Data

GPS data is embedded in a PVT (Position/Velocity/Time) frame fetched periodically from the microcontroller’s built-in GPS/LTE modem.
Vehicle Diagnostic Data

Diagnostic data is extracted from the car via the OBD-II diagnostic port.
Vehicle Diagnostic Data

Diagnostic data is extracted from the car via the OBD-II diagnostic port.
Vehicle Diagnostic Data

Diagnostic data is extracted from the car via the OBD-II diagnostic port

Data includes:
Vehicle Diagnostic Data

Diagnostic data is extracted from the car via the OBD-II diagnostic port

Data includes:
- Speed
- Engine RPM
Vehicle Diagnostic Data

Diagnostic data is extracted from the car via the OBD-II diagnostic port

Data includes:
- Speed
- Engine RPM
- Fuel Level
- Engine Load
Vehicle Diagnostic Data

Diagnostic data is extracted from the car via the OBD-II diagnostic port.

Data includes:
- Speed
- Engine RPM
- Fuel Level
- Engine Load
- Coolant Temperature
- Intake Air Temperature
Uploading Data to the Cloud
Uploading Data

CAN Data: [0x11, 0x22, 0x33, 0x44, 0x55, 0x66]
Uploading Data

CAN Data: [0x11, 0x22, 0x33, 0x44, 0x55, 0x66]

GPS Data: [05/22, 5:00PM, 34.41, -119.84, ...]
Uploading Data

- CAN Data: [0x11, 0x22, 0x33, 0x44, 0x55, 0x66]
- GPS Data: [05/22, 5:00PM, 34.41, -119.84, ...]
- Accel/Gyro Data: [0, 0, 9.8] [0, 0, 0]
Uploading Data

UDP Datagram

- **CAN Data:** [0x11, 0x22, 0x33, 0x44, 0x55, 0x66]
- **GPS Data:** [05/22, 5:00PM, 34.41, -119.84, ...]
- **Accel/Gyro Data:** [0, 0, 9.8] [0, 0, 0]
Uploading Data

GPS/LTE Modem

30s

UDP Datagram

CAN Data: [0x11, 0x22, 0x33, 0x44, 0x55, 0x66]

GPS Data: [05/22, 5:00PM, 34.41, -119.84, ...]

Accel/Gyro Data: [0, 0, 9.8] [0, 0, 0]
Processing Data in the Cloud
Processing and Persisting Data

UDP Listener (Port 6543)

Amazon EC2

UDP Datagram
- CAN Data: [0x11, 0x22, 0x33, 0x44, 0x55, 0x66]
- GPS Data: [05/22, 5:00PM, 34.41, -119.84, ...]
- Accel/Gyro Data: [0.6, 19]
Processing and Persisting Data

**UDP Listener (Port 6543)**

Amazon EC2

**Processed Data**

- **CAN Data:** [6 mph, 200 rpm, 14.22%, 44°F, 55s, 65°C]
- **GPS Data:** [34.4094°N, 119.8434°W]
- **Accel/Gyro Data:** [2 m/s², 0 m/s²]
Processing and Persisting Data

**UDP Listener (Port 6543)**

Amazon EC2

Processed Data

**CAN Data:** [6 mph, 200 rpm, 14.22%, 44°F, 55s, 65°C]

**GPS Data:** [34.4094°N, 119.8434°W]

**Accel/Gyro Data:** [2 m/s², 0 m/s²]

Amazon RDS
Serving Data to the User
Serving Data via API
Serving Data via API

Processed Data

CAN Data: [6 mph, 200 rpm, 14.22%, 44°F, 55s, 65°C]

GPS Data: [194.2312°N, 23.4534°W]

Accel/Gyro Data: [2 m/s^2, 0 m/s^2]
Serving Data via API

- CAN Data: [6 mph, 200 rpm, 14.22%, 44°F, 55s, 65°C]
- GPS Data: [194.2312°N, 23.4534°W]
- Accel/Gyro Data: [2 m/s^2, 0 m/s^2]
Serving Data via API

Processed Data

- CAN Data: [6 mph, 200 rpm, 14.22%, 44°F, 55s, 65°C]
- GPS Data: [194.2312°N, 23.4534°W]
- Accel/Gyro Data: [2 m/s², 0 m/s²]
Serving Data to the Front-End

Processed Data
- CAN Data: [6 mph, 200 rpm, 14.22%, 44 °F, 55s, 65 °C]
- GPS Data: [194.2312°N, 23.4534°W]
- Accel/Gyro Data: [2 m/s², 0 m/s²]

api.datadrivenucsb.com
Serving Data to the Front-End

API: api.datadrivenucsb.com
Serving Data to the Front-End

CAN Data: [6 mph, 200 rpm, 14.22%, 44°F, 55s, 65°C]

GPS Data: [194.2312°N, 23.4534°W]

Accel/Gyro Data: [2 m/s², 0 m/s²]

datadrivenucsb.com

api.datadrivenucsb.com

API

Amazon EC2

Processed Data

Amazon RDS

Map

FastAPI
Back-end Data Processing & Serving

Relational Database

<table>
<thead>
<tr>
<th>t</th>
<th>speed</th>
<th>temp</th>
<th>lat</th>
<th>long</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>23</td>
<td>45</td>
<td>67</td>
<td>89</td>
</tr>
</tbody>
</table>

UDP Listener

API

LTE-M

UDP Datagram

CAN data

GPS data

IMU data

nrf9160 SoC

Arm Cortex-M33

LTE Modem

GNSS Receiver

Vehicle Interface Subsystem

MCP2515 CAN Controller

TCAN330G CAN Transceiver

DB9 Connector

Tracker Module

Web-App Hosting & Authentication

data.driven.ucsb.com

amazon.com

api.datadriven.ucsb.com

UDP Listener 65432

API 8000

Data Driven

GPS
Web App
Web App: Live Tracking
Web App: Live Tracking

- **Track** vehicles live on an interactive map
Web App: Live Tracking

- **Track** vehicles live on an interactive map
- **Monitor** vehicle parameters live by selecting individual vehicles on the map
Web App: Live Tracking

- **Track** vehicles live on an interactive map
- **Monitor** vehicle parameters live by selecting individual vehicles on the map
- **Search** for parameters available from the dynamically populated sidebar
Web App: Data Visualization

![Data Visualization Web App](image-url)
Web App: Data Visualization

- **Plot** vehicle data parameters
Web App: Data Visualization

- **Plot** vehicle data parameters
- Supports selecting **multiple vehicles** for **comparing** data parameters in a date range
PCB Design
Schematics

μController & Sensor Peripherals
Schematics
Schematics

- nRF9160 SoC
- GNSS SMA
- LTE SMA
- MCP 2515
- TCAN 330G
- LSM6DSL IMU
- OBD-II
- Micro USB
- Bench Supply
- Sim Tray
- Micro USB
- MCU connector and USB drivers
- nRF9160 SoC
- Arm Cortex-M33
- LTE Modem
- GNSS Receiver
- Vehicle Interface Subsystem
- MCP2515 CAN Controller
- TCAN330G CAN Transceiver
- DB9 Connector
- LSM6DSL IMU
- Tracker Module
Layout
Layout

4-layer PCB

- Top Layer
- GND
- PWR
- Bottom Layer

131 components
Layout

4-layer PCB

- Top Layer
- GND
- PWR
- Bottom Layer

131 components
3D Rendering
Final Product

Few modifications made between PCB being sent out and rendering 3D model were left out.
Design Challenges

- RF circuitry is very sensitive
Design Challenges

- RF circuitry is **very sensitive**
- Learning & applying **best practices** on the fly
Design Challenges

- RF circuitry is **very sensitive**
- Learning & applying **best practices** on the fly
- Keeping track of **dynamic** requirements
  - Reworked GNSS & LTE RF circuitry from **onboard** antenna to **external** antennas
Design Challenges

- RF circuitry is very sensitive
- Learning & applying best practices on the fly
- Keeping track of dynamic requirements
  - Reworked GNSS & LTE RF circuitry from onboard antenna to external antennas
  - Accelerometer went out of stock
Design Challenges

- RF circuitry is **very sensitive**
- Learning & applying **best practices** on the fly
- Keeping track of **dynamic** requirements
  - Reworked GNSS & LTE RF circuitry from **onboard** antenna to **external** antennas
  - Accelerometer went **out of stock**
  - MicroSIM on nRF reference board was **cost-prohibitive**, so we pivoted to Nano SIM
Design Challenges

• RF circuitry is very sensitive
• Learning & applying best practices on the fly
• Keeping track of dynamic requirements
  ○ Reworked GNSS & LTE RF circuitry from onboard antenna to external antennas
  ○ Accelerometer went out of stock
  ○ MicroSIM on nRF reference board was cost-prohibitive, so we pivoted to Nano SIM
• Coordinating efforts within the team (not easily parallelizable)
Live Tracking Demo
Retrospective

- Modular design made it easy to develop each part of the system independently
Retrospective

- Modular design made it easy to develop each part of the system independently
  - Web App
  - API
  - UDP Listener
  - Tracker
Retrospective

- Modular design made it easy to develop each part of the system independently
  - Web App
  - API
  - UDP Listener
  - Tracker
- Learned a lot in every level of the technology stack:
  Web to Firmware to PCB design
Retrospective

- Modular design made it easy to develop each part of the system independently
  - Web App
  - API
  - UDP Listener
  - Tracker
- Learned a lot in every level of the technology stack:
  Web to Firmware to PCB design
- Got working PCBs in the **first spin**
  - On-board RF added significant complexity to the design process
Retrospective (cont.)

● More **test points**
Retrospective (cont.)

- More **test points**
- **Better labels** for non-power TPs, like SPI, I2C, etc. signals
Retrospective (cont.)

- More **test points**
- **Better labels** for non-power TPs, like SPI, I2C, etc. signals
- One **incorrect resistor** used for OBD-II power supply, soldered an external resistor
Retrospective (cont.)

- More **test points**
- **Better labels** for non-power TPs, like SPI, I2C, etc. signals
- One **incorrect resistor** used for OBD-II power supply, soldered an external resistor
- One board didn't have functioning UART, cause is TBD
Retrospective (cont.)

- More **test points**
- **Better labels** for non-power TPs, like SPI, I2C, etc. signals
- One **incorrect resistor** used for OBD-II power supply, soldered an external resistor
- One board didn't have functioning UART, cause is TBD
- Would've used separate LTE and GPS modem so we could use **streaming protocols** like MQTT
Next Steps
Acknowledgements

Dr. Yogananda Isukapalli
Alex Lai
Jimmy Kraemer
Venkat Krishnan

Brian Canty
Stefan Crigler
Martin Fay
Duane Gardner

Austin Hwang
David McCarthy
Eric Nystrom
John Buckley