**Logical View**

Location Processing  
  
UI (User Interface)

Trolley  
Location

Cloud Processing

Trolley Location App

**Process View**

ESP32 Data

Device

UI

Cloud Processing

Trolley Location

**Development View:**

Components:

ESP32 microcontroller connected to Wi-Fi

Cloud server or serverless backend

Mobile app for "Where's My Trolley"

SiriKit (for iOS) or Google Assistant integration

Web server and website for trolley location data and arrival time

Modules:

ESP32 firmware for collecting and transmitting trolley location data to the cloud server

Cloud server backend for receiving and processing trolley location data requests from the app and Siri/Google Assistant shortcuts

Mobile app for displaying trolley location data and approximate arrival time to the user

SiriKit (for iOS) or Google Assistant integration for handling voice commands and sending requests to the cloud server

Web server and website for displaying trolley location data and approximate arrival time to the user

Tools:

Visual Studio Code or other integrated development environment

Arduino IDE for ESP32 firmware development

Node.js for cloud server development

React Native or other mobile app development framework

SiriKit or Google Assistant SDK for voice assistant integration

HTML, CSS, and JavaScript for website development

Web server/website

Dislays trolley location data and approximate arrival time to users

Cloud server

Receives and processes trolley location data requests from the app and Siri/GoogleAssitant shortcut. Calculates trolley location and approximate time based on user location.

Mobile app

Displays trolley location data and approximate arrival time to the user. Sends requests to the cloud server for trolley location data and approximate arrival time based on user location.

Siri/Google Assistant

Processes voice commands from users and sends requests to the cloud server for trolley location data and approximate arrival time based on user location

ESP32 microcontroller

Collects trolley location data and transmits it to the cloud server

**Physical View:**

The physical view of the system consists of several physical components, including the ESP32 microcontroller, trolley sensors, Wi-Fi network, cloud server, mobile devices, and web server. The ESP32 microcontroller and trolley sensors are responsible for collecting location data and transmitting it over the Wi-Fi network to the cloud server.

The cloud server is responsible for processing and storing the location data received from the ESP32 microcontroller and providing access to this data to the other components of the system. The cloud server also hosts the backend APIs that are used by the mobile app and web server to interact with the system.

The mobile app and voice command shortcuts, such as Siri and Google Assistant, are responsible for requesting trolley location data and approximate arrival time from the cloud server based on the user's location. The mobile app and voice command shortcuts also provide a user interface for users to view trolley location data and approximate arrival time.

The web server is responsible for hosting a website that displays trolley location data and approximate arrival time to users. The web server also interacts with the cloud server through backend APIs to retrieve and display the trolley location data.

Overall, the physical view diagram and description show how the physical components of the system are connected and interact with one another to provide trolley location data and approximate arrival time to users.

ESP32

Wi-fi connected to trolley sensors

Cloud server

Receives trolley location data from ESP32 and processes requests

Mobile app

Requests trolley location data and approximate arrival time from cloud server based on user location

Siri/Google Assistant

Processes voice commands and requests trolley location data from cloud server

Web/server/website

Displays trolley location data and approximate arrival time to users

Scenarios:

Scenario 1:

1. User opens the app.
2. App displays a search bar and a "Find Trolley" button.
3. User enters the trolley number.
4. App sends the trolley number to the cloud server.
5. Cloud server receives the request and looks up the location data for the trolley.
6. Cloud server sends the location data back to the app.
7. App displays the trolley's location to the user.