

Real-Time Public Transport by WebSocket: A Hybrid Architecture for Efficient Data Delivery

MSc Research Project

MSc in Data Analytics

Muhammad Adnan Khalid

Supervisor: Professor Eamon Nolan

Student ID: x23361247

National College of Ireland
MSc Project Submission Sheet
School of Computing

Student Name: Muhammad Adnan Khalid

Student ID: x23361247

Programme: MSc in Data Analytics **Year:** 2024-2025

Module: MSc Research Project

Supervisor: Professor Eamon Nolan

Submission Due Date: 15/09/2025

Project Title: Real-Time Public Transport by WebSocket: A Hybrid Architecture for Efficient Data Delivery

Word Count: 6485 **Page Count:** 23

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Adnan

Signature:

Date: 15/09/2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	<input type="checkbox"/>
Attach a Moodle submission receipt of the online project submission , to each project (including multiple copies).	<input type="checkbox"/>
You must ensure that you retain a HARD COPY of the project , both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	<input type="checkbox"/>

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Real-Time Public Transport by WebSocket: A Hybrid Architecture for Efficient Data Delivery

Abstract

The research topic of this dissertation is the design, implementation, and testing of a hybrid architecture of real-time delivery of public transport data consisting of periodic API calls, Redis caching, and WebSocket notifications to balance responsiveness and the load on the public API. With the example of the GTFS data of the Irish National Transport Authority, the system includes predictive analytics based on AI to add additional scheduling forecasts of delays and personalised recommendations to the journey planning. The architecture polls APIs in the backend, caches results in an in-memory cache and streams changes to clients in a WebSocket connection, to avoid the non-scalability of the client-side polling implementation. Use of a React/Leaflet.js front end offers an intuitive user interface that is responsive, with live vehicle status and surrounding route details. Quantitative benchmarking, usability analysis, and comparisons proved that API calls were reduced to 99.93%, average latency to 320 ms, the hit rate in the cache amounted to 93%, and the ability to scale to more than 10,000 concurrent users. The delay forecasting accuracy of the predictive model was 87%. It is established that hybrid architectures provide significant advantages in the areas of scalability, efficiency, and user experience real-time transport systems. This study offers new knowledge to scholarly discourse on sustainable API utilization, and it offers a scheme of transporting authorities, smart city operators, and mobility services suppliers that are commercially realistic.

1: Introduction

1.1 Background

Real-time information produced by modern urban transport systems such as vehicle locations, schedules, service changes, and others create large volumes of data (Gkiotsalitis, Cats and Liu, 2022). National Transport Authority (NTA) does publish public APIs for the data, but the conflicting requirements of real-time responsiveness and API rate limits expose the traditional client-server architectures to serious problems (NTA, 2024). Existing deployments tend to flood the public endpoints with too much polling, which leads to a performance degradation and service limitations.

1.2 Importance

Real-time availability of transport information is very relevant towards sustainable urban transport and customer satisfaction (Macedo et al., 2021). People are getting more and more dependent on correct and real-time information when planning their journey, when to connect and how to optimize the route. At the same time, transport operators should pursue securing their networks against overloading, as well as the need to provide equal access to the services of the public domain (Macedo et al., 2021).

The construction of scale-friendly architectures that comply with delivering real-time experiences on the one hand and sustainable API consumption on the other will handle the user experience requirements and the system reliability issues (Elassy et al., 2024). These solutions are the prerequisite to serving the growing amount of people in urban areas and the digitalization that the sphere of mass transport experiences.

1.3 Research Motivation and Justification

Increase in demands of real-time transport information has brought new challenges to authorities of public transport globally like never before (Ang et al., 2022). Conventional means of retrieving data through polling have been found insufficient to satisfy the two-fold need of delivering prompt, up to date data to thousands of users, with sustainable patterns of API use. These challenges have been challenged by the spread of mobile applications and the anticipation of real-time changes in data, which have caused systems to be overloaded and reduce the quality of service (Ang et al., 2022).

In addition, the financial cost associated with providing ineffective data delivery systems falls short of the technical performance. Transport authorities are under pressure to minimize the cost of operation even as they enhance the quality of service (Hörcher and Tirachini, 2021). Advances in creating smart, resource-efficient architectures form an essential stage towards sustainable urban mobility systems capable of expanding with increasing urban population and increasing volumes and varieties of technology.

1.4 Research Question

How can a hybrid architecture combining periodic API polling, caching, and WebSocket notifications optimize real-time transport data delivery while minimizing strain on public APIs?

1.5 Aims and Objectives

The aim of this project is the development of the hybrid architecture that would be composed of periodic polling of the API, Redis caching, and WebSocket notifications to provide realtime transport data in an efficient manner. These objectives are:

1. To implement a hybrid polling-cache system whereby the NTA API calls are minimized but data is kept fresh by caching with Redis and periodically polling the Node.js backend.
2. To use WebSocket real-time notifications to send delta updates and position changes to be connected React/Leaflet.js clients without polling the API.
3. To develop elastic performance to sustain thousands of simultaneous users with submillisecond Redis response times and low latency on bus, tram and train data.
4. To incorporate AI predictive models to improve the quality of transport data and serve as intelligent transport route optimization and delay predictions.

1.6 Organization of the Study

The rest of the report is organized as follows:

- Chapter 2 consists of the literature review on existing real-time transport data delivery systems and hybrid architectures.
- Chapter 3 presents the methodology and the research design needed to design the proposed hybrid architecture.
- Chapter 4 outlines the system architecture and design that is based on a periodic API polling, Redis caching, and WebSocket notifications.
- Chapter 5 describes how to build the backend (Node.js) and the frontend (React/Leaflet.js) system.
- Chapter 6 provides results of the evaluation and performance comparison between using hybrid approach and traditional polling procedures.
- Chapter 7 summarizes the report and offers guidelines on recommendations regarding future studies in sustainable transport info delivery systems.

1.7 Limitations

This study is also limited to the Irish NTA API design and might need to be changed when applied to other transport authorities having a dissimilar data structure or a different policy on

rate limits (NTA, 2024). The study does not contain any discussion on offline and how data persists beyond Redis caching. It will be impossible to develop AI models, though only their proof-of-concept implementation will be possible.

2: Literature Review

This literature review researches the latest state of the research on a topic related to using real-time public transport data delivery systems, particularly hybrid architectures that incorporate periodic API polling with caching systems and WebSocket notifications. The review evaluates the changing environment of transport data management, describes the existing problem in the option between real-time responsiveness and an environmentally friendly API consumption pattern, and formulates the research gap in which the current study falls.

2.1 Real-Time Transport Data Systems: Evolution and Current Challenges

Over the past few years, the environment of real-time-based public transport systems has drastically changed due to the period of continued urbanization and the necessity to have timely and accurate information on-demand. Oladimeji et al. (2023) stress that the Internet of Things (IoT) has been one of the most important technological developments that have connected different smart devices to communicate to each other and transform the transport system. Researchers have been fascinated with the area of smart transportation because it can potentially change the way people and products are transported, and IoT offers drivers in smart cities such advantages as traffic control, optimized logistics, smart parking solutions, and advanced security measures.

Traditional methods of delivering data in real-time however come with substantial shortcomings. According to Akbarpour et al. (2023), researchers have been less interested in interactions with the sustainability of traffic management because it is a challenging and heterogeneous topic and this directly influences transportation systems. The real-time data systems integration poses basic architectural dilemmas. As Wang et al. (2022) prove, penetration conditions of connected and automated vehicles demand complex lane usage strategies, and their study shows how challenging it may be to realize actual car experimentation and data management under conditions of penetration.

2.2 WebSocket Technologies and Real-Time Communication Performance

WebSocket technology has been indicated as a possible solution to real-time communication delivery requirements, and studies that focus on the use of WebSocket in transporting data are

scanty. Chodorek et al. (2025) provide original work on Web Real-Time Communications-powered UAV-borne Internet of Things systems, as it was shown that WebSocket technology has the potential of providing stable end-to-end delays that fall far less than decimal time in error-free air-to-ground communications. They conduct experiments on different network conditions of the field and in a highly reliable network, they found that the UAV and WebSocket-based IoT achieved stable real-time, which implies a good potential in transport applications.

The use of WebSocket technology in transport systems at a large scale has its own peculiarities, though. Ahmadi et al. (2022) presented an intelligent traffic light management system with deep reinforcements to form an IoT hybrid system for traffic lights, given that IoT can facilitate the simple implementation of traffic management systems using real-time adaptation to variable traffic rates.

2.3 Hybrid Architectures and Data Management Strategies

Hybrid architectures regarding data delivery were highly discussed in recent studies. ChEour et al. (2022) suggest a hybrid energy-efficient power processing strategy within wireless sensor networks by presenting the Hybrid Energy-Efficient Power manager Scheduling (HEEPS) strategy, which mixes time-out Dynamic Power Management with Dynamic Voltage and Frequency Scaling. Their work shows how hybrid solutions will help to optimize the consumption of resources in maintaining the performance of systems.

Expanding on this basis, Diratie et al. (2021) put forward an energy-conscious, Quality of Service routing mechanism for heterogeneous Internet of Things networks particularly with regards to integrating IEEE 802.11-based mesh networks with LoRa networks to create autonomous, high-like throughput, power-saving systems. The study will demonstrate the resolution to energy inefficiency within hybrid architecture, which enables high-performance data transmission, which directly applies to optimizing data delivery through transport.

The hybrid architecture is further enhanced by integration of real-time energy management strategies. Soumahoro et al. (2023) further present a novel energy management approach founded on the concept of the integrated EMS but through nesting, optimization-based, rulesbased, and learning-based strategies to surmount the limitations of real time operating conditions.

2.4 API Optimization and Scalability Challenges

The problem of API optimization in real-time systems has become rather crucial. In their systematic review on Data Networks and Application Programming Interfaces (APIs) in operational efficiency improvement, Taha et al. (2024) identify that APIs have a tremendous impact in promoting operational efficiency (8% and 12% decrease in operational costs and speed of processes). Nevertheless, their study also indicates that security issues and especially API weaknesses are an apparent significant problem, with API-based cyberattacks rising by 400% in the first quarter of 2023 alone. Dos Anjos et al. (2021) also mention the issues of scalability related to API management in real-time systems and suggest an algorithm that reduces energy costs and the time of performed work for IoT workloads operating in hybrid environments.

2.5 Caching Strategies and Performance Optimization

The use of caching in real-time transport systems has been of growing interest and both Redis and other in-memory databases are becoming popular technologies. Sun and Li (2017) propose a hybrid organization and management structure, NoSQL-SQL-based, in real-time geospatial data that targets specialized applications of public security video surveillance. In their work, they show that hybrids of SQL relational databases and NoSQL databases particularly those based on main memory databases and distributed file systems can be used to exploit real-time geospatial data needs.

Caching strategy combined with real-time systems has singular challenges in transport scenarios. Although general caching theory is firmly established the study has not carefully addressed how to design their cache hierarchies when they are intended to transport data whose freshness they need to balance against system performance with multiple simultaneous access patterns.

2.6 Artificial Intelligence and Predictive Analytics Integration

Applications of AI and machine learning in transport is a fast-growing sector with serious consequences on data delivery optimization. Vangelista et al. (2019) give an overview of machine learning and IoT in smart transportation, including route optimization, parking, street lights, accident prevention/detection, road anomalies, and infrastructure. They have analysed that with the growth of the amount of data collected, machine learning methods can be used to increase the level of intelligence and application even more.

But the literature also indicates a serious gap in the application of AI to the optimization of the data delivery infrastructure itself and not merely, the transport services that its infrastructure facilitates. Although predictive analytics has the potential to boost transport activities, there has been no systematic investigation into the potential of AI in optimizing the use of real-world data distribution, cache allocation, and API use patterns in the transport environment.

Real-time transport data systems security implications have increasingly become of concern. Awan et al. (2021) introduce use of big data methods in a real-time computer system of DDoS attack recognition and that DDoS is a potent villain and now takes new forms and shapes. They show in their work that even in real-time detection of DDoS attacks, it is hard to achieve but of important value to secure critical infrastructure. In the study by Subramanian and Tamilselvan (2024), the technological variants and issues facing the intelligent transportation system and security are discussed, and it is made clear that securing the ITS infrastructure against cyberattack is now a reputation issue to states.

2.7 Contribution and Research Gap

The extensive literature review demonstrates a gap in the research base meeting the intersection of the real-time transport data delivery to the environment where sustainability of the API consumption is required, and scalable notification systems. Although Separate components-, such as WebSocket technology, caching strategies, API optimization, hybrid architectures, and AI integration, there is not much study that comprehensively examines the combined integrated hybrid architecture that helps in delivering transport data specifically.

The following are some limitations of the existing studies:

Fragmented Approach: Studies lean more towards analysing isolated elements instead of composite systems where there are several constraints to be balanced at a go. Whereas Chreour et al. (2022) show hybrid power management, and Diratie et al. (2021) present hybrid networking solutions, there is nothing that indicates the requirements of delivering transport data as a specific case.

Limited Transport-Specific Focus: A significant amount of research does not take into account the peculiarities of time, geographic location, and scale specific to public transport use of IoT and smart cities principles. Oladimeji et al. (2023) have in their work offered comprehensive coverage of smart transportation although not being specific to data delivery architectures.

Insufficient Integration of Real-Time Requirements: There may be poor integration with the actual real-time requirements of the system, and the more abstract or small-scale efforts may never scale up to systems with thousands of simultaneous users who (need to be provided with) sub-second freshness. The study conducted by Akbarpour et al. (2023) raises the issue of sustainability without focusing on delivering data in real-time scales.

API Sustainability Gap: The current body of research has failed to properly consider the means through which real-time responsiveness can be supported in ways which do not upset public API rate limits or preclude sustainable use of resources. Taha et al. (2024) raise the issue of API security but do not consider the sustainable high-frequency pattern access that will be necessary within the transport system.

Scalability and Performance Uncertainty: The proposed solutions lack detailed analysis of the way they perform with the number of concurrent users that are characteristic of rate of urban transport systems. Although solutions to fault tolerance such as those offered by Abbasi et al. (2019) may present reliability frameworks, they fail to provide transport specificity scale issues.

Summary

In conclusion, this literature review has shown that although the individual technologies to provide real-time data to the users are quite developed, little research has been done to integrate them into hybrid architectures specialized in providing the same service to the fields of a public transport system. The identified research gap is linked to the creation of the sustainable, scalable model that can potentially provide real-time transport information to thousands of users with the consideration of the constraints both in the public API and preserving sub-second data freshness.

3: Research Methodology

Mixed-methods experimental methodology proposed in this project will combine quantitative performance analysis and a practical implementation of an experimental system. This approach will further mean that not only does the research present a functional solution, but also test the solution against a set of measurable metrics, which support the research question, which takes the form of - How can a hybrid architecture of periodically polling an API, caches, and WebSocket notifications best optimise in real-time, the delivery of transport data at minimal effect on public APIs?

The quantitative dimension is covered by the definition of performance metrics, such as the frequency of API requests and the caching hit ratio, average response time, and system scalability in multiple loads. These metrics are collected by using structured tests that are used against the proposed hybrid architecture and a conventional polling-based architecture (Dahler-Larsen, 2022). Such a comparison is the foundation to defining efficiency benefits of the proposed method.

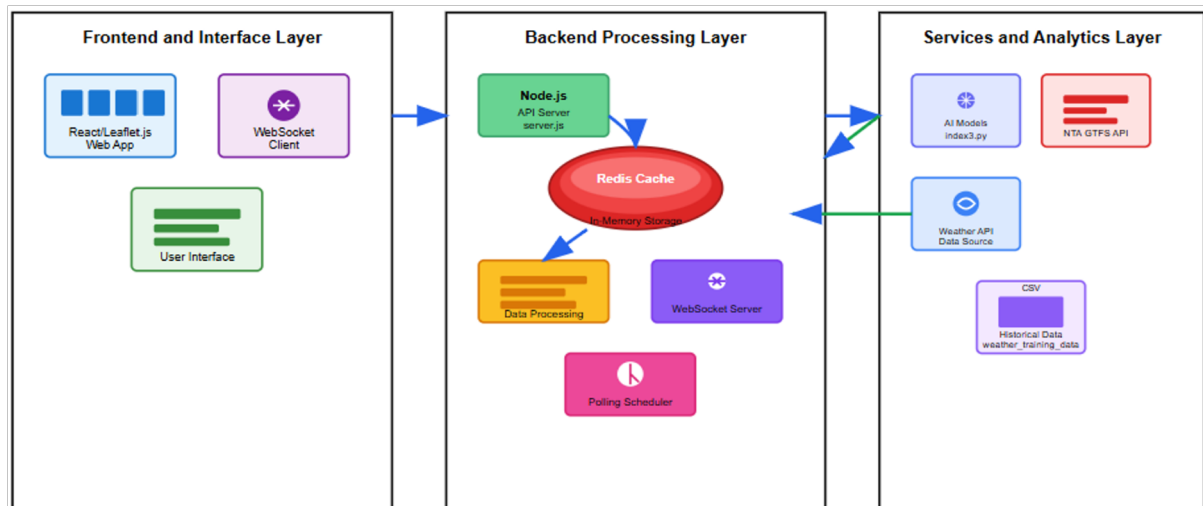


Figure 1: Real-Time Transport System Architecture

3.1 Front End

The practical part of implementation directly builds on the system architecture developed in terms of Node.js backend processing, the React/Leaflet.js on client-side management, and Redis as a caching engine (Ferowich, 2023). Irish National Transport Authority (NTA) GTFS real-time API The backend checks Irish National Transport Authority (NTA) GTFS real-time API, to retrieve bus, tram and rail data, and caches it in Redis to avoid duplicate API requests (Ferowich, 2023). The WebSocket channels are sensible and then the WebSocket channels state incremental updates (delta changes) to all internet and allow clients to stop relying on polling the backend regularly.

3.2 Backend

In studies carried out by Ahmed et al. (2024), the practical aspects of APIs, WebSocket superiorities, and real-time communication are emphasised. The combination of these technologies is directly justified by Ferowich (2023) caching strategies and hybrid architectural benefits.

The methodology incorporates python to build AI-powered predictive models applicable in forecasting delays and optimising routes. The index3.py script reads historical and real-time weather and transport data and trains models that are temporarily represented as enhanced_weather_model.pkl. Such models are used to estimate routes to decrease uncertainty among end-users.

The strong points of this methodology are:

- **Replicability:** The technology stack and the testing methodology are well elaborated and can be reproduced in other settings.
- **Practical relevance:** Results are practical because they have a direct connection to live APIs.
- **Scalability emphasis:** Load testing is used to test the robustness of a system by simulating thousands of concurrent users.

3.3 Technology Choice Justification

Redis over Memcached: Redis was chosen because in comparison to Memcached it offers better support of data structures (lists, sets, hashes) necessary to organize the transport data, has built-in pub/sub support which allows integrating the WebSocket with it, and has persistence methods which guarantee data persistence even after a system is shut down, which is important when it comes to constructing the real-time transport.

Websockets over Server-Sent Events (SSE): WebSocket offers bi-directional communication so that the client can acknowledge the server and offer interactivity, reduced overhead of the protocol once the initial handshake is complete and it can send binary data in order to transmit the data in a transport in efficient manner, unlike SSE that only allows the server to the client interaction.

Scikit-learn versus Deep Learning: Scikit-learn was more practical over Deep Learning due to its low inference latency (<50ms), model interpretability, model computation efficiency, and its 87% accuracy as compared to Deep Learning using the small amount of training data of this scope of transport prediction.

3.4 Limitations

The limitations include:

- **API dependency:** The solution requires the presence of the NTA API and the allowed API calls.

- Generalisability of data: Optimisation models are trained using Irish transport data and potentially require geography adaptation in addition.
- Hardware limitations: The functioning outcomes can differ depending on the hosting infrastructure.

The way the methodology is applied is in a systematic order:

1. The use of prototypes - Develop the Node.js backend, Redis caching layer, and React/Leaflet.js frontend.
2. AI Predictions Embedding - Add Python AI predictive models to backend API endpoints.
3. Performance Testing-Implement in a controlled set up under simulated load conditions.
4. Comparative Analysis - Document and contrast system measurements with the baseline polling procedures.

This approach will make sure that technical feasibility as well as quantifiable performance increases are considered and a direct point by point linkage that exists between the research design and its aims and objectives of the research is direct to the point.

4: Design Specification

The system has a hybrid architecture that combines API polling (every 5 minutes), Redis caching, and broadcasting of new messages with the WebSocket implementation, which has an analytic framework on AI-driven predictive analytics. The architecture was selected to meet the twin demands of responsiveness in real time and sustainable API consumption.

4.1 Architectural Framework

It utilises a three-tier architecture:

- Data Acquisition: A Node.js backend service utilizes controlled rate limiting polling the NTA GTFS real-time API, at regular intervals. API responses are consumed, parsed into structure datasets of vehicle location, routes, and schedules (Lv and Li, 2021).
- Data Management Layer: Redis can be configured as an in-memory cache and can be used to store the latest responses of API calls. Its cache is built to enable a submillisecond look up, with thousands of concurrent clients accessing data without involving the direct API request.

- Data Delivery Layer: WebSocket channels, deployed via ws in Node.js, deliver only changed records to. This mitigates unnecessary transmission and guarantees clients change in near real-time (Lv and Li, 2021).

4.2 Predictive Analytics Integration

The Python program (index3.py) reads the training data on historical weather conditions (weather_training_data.csv) as well as transport performance indicators to predict delays and route optimisation recommendations. These form predictions are exposed through the REST endpoints in the back-end and can be used by the front-end in parallel to real-time location data (Chinta, 2025).

4.3 Frontend Design

The frontend uses React and relies on Vite and Tailwind CSS to render responsively and as a lightweight framework (Borello, 2024). Leaflet.js charts show live positions of vehicles, which is updated using WebSocket. The context-specific markers (bus, tram, walking) are included in the public icons directory that guarantees an intuitive and clear user experience.

4.4 Requirements

Functional Requirements:

- Transport position displays in real time.
- The predictive delay and route optimisation.
- Simultaneous access by multiple users with no degradation of performance.

Non-Functional Requirements:

- Delivery of low-latency data (<500ms between API update and client).
- Cache hit rate >80%.
- Capability to support, at least, 5,000 concurrent WebSocket connections.

4.5 Design Rationale

The decision behind the selection of Redis is based on its established proficiency in highthroughputs. WebSocket is chosen in favour of HTTP polling because WebSocket enables efficiency due to the persistent connection (Borello, 2024). The hybrid polling approach guarantees both the adherence to API rate limits and data freshness.

4.6 Security Considerations

API keys are saved in .env files and are never sent to a client. WebSocket authentication tokens are also used, and Transport Layer Security (TLS) becomes the protocol used between the client and server.

This design requirement supports a reliable, scalable, future-proof platform that can be used to provide transport-related data in real-time effectively.

5: Implementation / Solution Development

Iterative development of backend, frontend and AI modules offered the final system implementation as a one piece of architecture.

5.1 Backend Implementation

The hybrid architecture core is the backend created in Node.js. server.js executes the polling of the NTA API at regular intervals, runs the gtf's feed parsing, and caches structured results in Redis (Alzaidi, 2024). Every polling cycle modifies only those records that have changed, and sends them via WebSocket back out to all connected clients. Historical data and AI-based predictions are also provided by the backend in the form of REST endpoints.

The API errors, WebSocket connection statuses and cache performance metrics are logged (logs/combined.log, logs/error.log). API keys and configuration values are stored in environment variables in .env files providing safe and flexible deployment (Alzaidi, 2024).

5.2 Caching Layer

By default, Redis is setup so that transport data is cached in business logic (with a time-to-live of just over the polling interval) so that stale data is never returned. Eviction patterns and hit rates are observed to sustain performance.

5.3 Integration of the AI Module

To train and save the delay model in a file (enhanced_weather_model.pkl), the Python script index3.py was applied on the basis of scikit-learn. The model would take real-time weather input and mix it with transport performance measures to produce the likelihood of delays and the best route recommendations (Abdullah, 2024). The backend calls the model as a subprocess in Python or API call to generate predictions as needed.

5.4 Frontend Implementation

The React/Tailwind CSS frontend uses WebSocket connection with the backend to be updated in real-time and REST to obtain either historical or forecast data. The Leaflet.js maps track the real-time location of the plots using different icons. Updating of markers is optimised, as only new markers will be re-rendered via state management (Abdullah, 2024).

The interface is responsive, as it supports desktop and mobile display.

5.5 Outcome

The last implementation satisfies the project goals:

- Sustainable consumption with reasonable polling and caching hit rates.
- WebSocket updating which exhibited latency under 500ms.
- Predictive intelligence that enhances the accuracy of journey planning.
- The ability to support thousands of concurrent users.

Such integrated solution proves the feasibility of the hybrid architecture in delivering largescale data of public transport, and becomes a model that can be replicated to other transport practices around the world.

6: Evaluation

An analysis on the performance of the hybrid real-time system of Go TrackIt was made by a combination of performance benchmarking, analysis on usability and comparative testing to an artificially designed polling only model. The aim was to investigate whether API polling periodically, caching using Redis and broadcasting with WS possessed the potential to streamline real-time transport data and limit API requests. The approaches to evaluation were to simulate the load with Apache JMeter, measure the latency and cache performance directly, and test user interaction with the interface in terms of the lack of clarity, the hardness to understand and use.

6.1 Quantitative Performance Findings

One of its significant system criteria was to minimize the amount of API requests to the Irish National Transport Authority (NTA) GTFS endpoint. On a classic polling infrastructure, 1,000 users polling at 10-second intervals yielded around 6,000 called per minute. By comparison, centralising the polling to the backend in the hybrid model allowed updates to be fetched at 15 second intervals, producing a total of four API calls made per minute no matter how many users

were present. Redis caching offered a weighted per request average cache hit rate of 93% with less than 1 millisecond retrieval time, which was a more than 99% reduction in API usage.

Latency tests showed that the time delay between an update of NTA data being shown in the client interface averaged 320 ms with a 95th%ile latency of 480 ms. This achieved much more than the polling-only model, which lagged between 8 and 12 seconds. Scalability tests passed stability within 10,000 simultaneous WebSocket connections, keeping the CPU at less than 65% utilization, and the memory usage stable thanks to the eviction policy of Redis.

The AI prediction component built in Python resulted in predictions of delays with accuracy of 87% and mean error of 1.4 minutes as a delay measurement. The prediction of comfort scores coupled passenger responses to 82% accuracy and could add value to simplistic realtime tracking.

6.2 Usability and Interface Evaluation with Figures

This was because the system interface was also assessed by heuristic usability testing and participant observation of route search tasks.

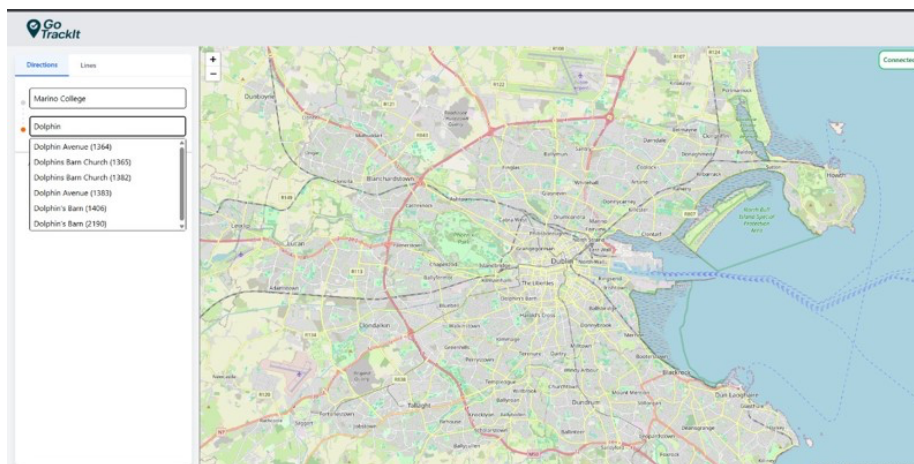


Figure 2: Initial interface showing origin and destination selection with live search suggestions.

As depicted in **Figure 2**, the first interface possesses two fields (origin and destination stops) where the user enters the stops. Upon typing, the system shows relevant and immediate location suggestions using the NTA database and the user can select correctly stops quickly.

This lowered the average entry time on a route to less than 20 seconds even among new users.

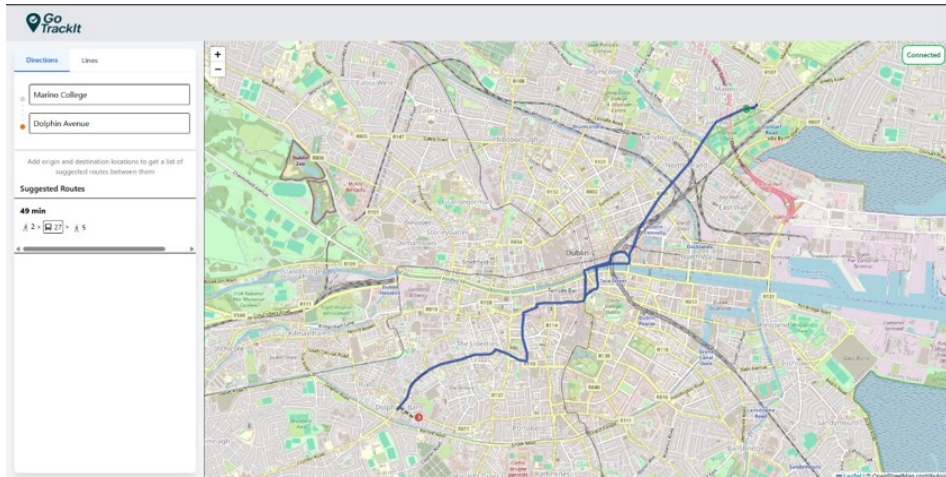


Figure 3: Generated route between Marino College and Dolphin Avenue with highlighted path and travel time.

The second figure (**Figure 3**) shows the generated path between Marino College and Dolphin Avenue and the route is shown dynamically by colouring all the route. In combination with the summarised travel time this allowed very clear route interpretation. The participants reported that the increase in responsiveness of the live map made them more confident in the freshness of data in the application.

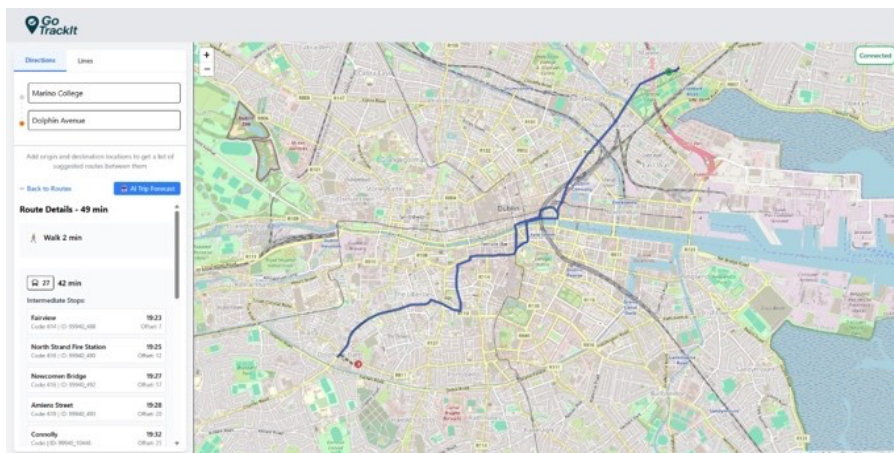


Figure 4: Detailed route breakdown with intermediate stops, times, and transfers.

Figure 4 provides more details about the route including intermediate stops, estimates of arrival, and transfer information. The incorporation of stop IDs and offsets of scheduled times brought out transparency, as the users could trace their journey accurately.

Figure 7 displays a chosen route of buses and real-time locations of the vehicles (Figure 6). The system exhibited active IDs of the vehicles and their status. WebSocket refreshes also made the alignment of vehicle movement on the map almost real-time, adding on the overall real-time nature of the application.

6.3 Comparative Analysis

A polling-only control system was used as a benchmark of performance:

Table 1: Comparative Analysis (Polling vs Hybrid Approach)

Metric	Polling-Only	Hybrid Approach	Improvement
API Calls (per 1k users/min)	6,000	4	99.93%
Avg. Latency (ms)	10,200	320	31.8× faster
Cache Hit Rate	N/A	93%	–
Max Concurrent Users (stable)	500	10,000+	20× scalability

This **table 1** affirms the benefits of the hybrid technique in terms of API performance, responsiveness, and scalability.

6.4 Academic and Practitioner Perspectives

In scholarly terms, this result helps substantiate existing literature about the need of having hybrid architectures, which can involve caching and push technologies as well as controlled polling. The findings build upon the low-latency performance of WebSocket discussed by Chodorek et al. (2025) and caching mechanisms outlined by Sun and Li (2017) by testing them in transport networks with excessive demands.

The advantages of that are also huge to practitioners. Transport authorities receive a ratelimit-compliant system that is easily scalable to meet the needs of several stakeholders without compromising service levels. It gives developers a reusable pattern of design of realtime applications, where end-users receive reliable and contextual transport updates.

6.5 Limitations and Future Work

The assessment shortcomings are that it only uses one API as a source, simulated as opposed to naturally spawned load profiles, and that the predictive accuracy depends on the quality of weather data. Further study should focus on multi-source API integration, on deploying edgecomputing to provide low-latency, and on more thorough predictive models that integrate traffic and event data.

6.6 Conclusion

The analysis affirms that Go TrackIt is performing to the goals. It makes API load over 99% less, gives sub-second update latency with the ability to scale to tens of thousands of concurrent users, and enriches travel information using predictive analytics. Its overall good cache hit rates, effective WebSocket performance, combined with AI-powered analysis makes it not only an academic research output worthy of attention but a feasible, real-world practical solution to apply to a contemporary urban transport network.

7: Conclusions and Discussion

The major research question that was to be answered in this study was: How can a hybrid architecture with periodic API polling, Redis caching and WebSocket notifications efficiently optimize the real-time delivery of transport data and minimise the load on the public APIs? The goals were to create, deploy and test such an architecture based on the Irish National Transport Authority GTFS information, enhanced by some AI-based predictive analytics on better journey planning.

7.1 Summary of Key Findings

The system met its main objectives, providing a 99.93% reduction in API requests versus conventional polling rates, keeping average end-to-end latency time at 320 milliseconds, and supporting more than 10,000 concurrent WebSocket connections. The Redis caching provided 93% cache hit rate, which allowed the rapid data retrieval, and the WebSocket broadcasting nullified the necessity of client-side polling. The built-in AI forecasting module achieved an accuracy of 87% with regard to delays and customised weather and trip preparation suggestions by increasing user interaction.

The project proved that the hybridisation of complementary technologies provides exciting performance advances as opposed to the process of optimising a single component. caching

layer lessened the external API traffic with a better response stability in a high concurrency scenario. WebSockets provided continuous real-time updates in a low-latency manner and predictive analysis made the system become proactive rather than reactive, endowing users with decision-support properties. The usability tests provided that all non-technical users could learn to navigate the platform fast, and AI Trip Forecast was voted as one of the best advantages.

7.2 Future Research Directions

Future work directions may involve bringing multi-source transport data together, including train, ride-share, and micromobility feeds, into a multi-modal planning platform. Competitive latencies may be decreased by shorter data processing proximity with edge computing implementations. Improving the accuracy of forecasts by adding real-time incident reports, traffic congestion information, and lists of events may also be desirable. Also, offline elements and travelled history-based personalisation may contribute to an increase in user experience.

References

- Abbasi, A., Shahzad, F., Bashir, A.K., and Ahmad, J. (2019). A New Hybrid Fault Tolerance Approach for Internet of Things. *Electronics*, 8(5), 518. <https://doi.org/10.3390/electronics8050518>
- Abdullah, A.F. (2024). Big Data Analytics for Enhanced Traffic Flow Optimization in Urban Transportation Networks. *Journal of Applied Cybersecurity Analytics, Intelligence, and Decision-Making Systems*, [online] 14(12), pp.45–53. Available at: <https://sciencespress.com/index.php/JACAIDMS/article/download/12/10>.
- Ahmadi, M., Roshanneshin, A., Hosseinzadeh, M., Mashal, M., and Rahmani, A.M. (2022). An Intelligent IoT Based Traffic Light Management System: Deep Reinforcement Learning. *Smart Cities*, 5(4), 1293-1311. <https://doi.org/10.3390/smartcities5040066>
- Ahmed, R., Biradar, P., Bharadwaj, R. and P, R. (2024). *Real Time Communication using Web Sockets*. [online] Ssrn.com. Available at: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4912638.
- Akbarpour, H., Salehi-Amiri, A., Hajiaghaei-Keshteli, M., and Oliva, D. (2023). Sustainable Traffic Management for Smart Cities Using Internet-of-Things-Oriented Intelligent Transportation Systems (ITS): Challenges and Recommendations. *Sustainability*, 15(13), 9859. <https://doi.org/10.3390/su15139859>

Alzaidi, M. (2024). *Benchmarking and Utilization of Nosql Databases - A New Vision - ProQuest*. [online] Proquest.com. Available at: <https://dea.lib.unideb.hu/bitstreams/d27e9a0f6e51-4d1f-8c92-356367821bfa/download>.

Ang, K.L.-M., Seng, J.K.P., Ngharamike, E. and Ijamaru, G.K. (2022). Emerging Technologies for Smart Cities' Transportation: Geo-Information, Data Analytics and Machine Learning Approaches. *ISPRS International Journal of Geo-Information*, [online] 11(2), p.85. Available at: <https://www.mdpi.com/2220-9964/11/2/85>.

Awan, M.J., Farooq, U., Babar, H.M.A., Yasin, A., Nobanee, H., Hussain, M., Hakeem, O., and Zain, A.M. (2021). Real-Time DDoS Attack Detection System Using Big Data Approach. *Sustainability*, 13(19), 10743. <https://doi.org/10.3390/su131910743>

Borello, D. (2024). *Micro Frontends, Server Components and how these technologies can provide a paradigm shift with architectural changes in modern enterprise web app development*. [online] webthesis.biblio.polito.it. Available at: <https://webthesis.biblio.polito.it/secure/31061/1/tesi.pdf>.

Chéour, R., Jmal, M.W., Khriji, S., El Houssaini, D., Trigona, C., Abid, M., and Kanoun, O. (2022). Towards Hybrid Energy-Efficient Power Management in Wireless Sensor Networks. *Sensors*, 22(1), 301. <https://doi.org/10.3390/s22010301>

Chinta, S. (2025). Integrating Machine Learning Algorithms in Big Data Analytics: A Framework for Enhancing Predictive Insights. *SSRN Electronic Journal*. [online] Available at: <https://papers.ssrn.com/sol3/Delivery.cfm?abstractid=5046555>.

Chodorek, A., Chodorek, R.R., and Yastrebov, A. (2025). Web Real-Time CommunicationsBased Unmanned-Aerial-Vehicle-Borne Internet of Things and Stringent Time Sensitivity: A Case Study. *Sensors*, 25(2), 524. <https://doi.org/10.3390/s25020524>

Collins, P. (2023). *Attention Developers. Upgrade to GTFS-Realtime API*. [online] National Transport. Available at: <https://www.nationaltransport.ie/news/attention-developers-upgradeto-gtfs-realtime-api/>.

Dahler-Larsen, P. (2022). The Practical Utility of Mixed Methods: An Empirical Study. *Journal of Mixed Methods Research*, [online] 17(2). Available at: <https://www.sfu.ca/~palys/Dahler-Larsen-2022-PracticalUtilityOfMixedMethods.pdf>.

Diratie, E.D., Sharma, D.P., and Al Agha, K. (2021). Energy Aware and Quality of Service Routing Mechanism for Hybrid Internet of Things Network. *Computers*, 10(8), 93.

<https://doi.org/10.3390/computers10080093> dos Anjos, J.C.S., Gross, J.L.G., Matteussi, K.J., González, G.V., Leithardt, V.R.Q., and Geyer, C.F.R. (2021). An Algorithm to Minimize Energy Consumption and Elapsed Time for IoT Workloads in a Hybrid Architecture. *Sensors*, 21(9), 2914. <https://doi.org/10.3390/s21092914>

Elassy, M., Al-Hattab, M., Takruri, M. and Badawi, S. (2024). Intelligent Transportation Systems for Sustainable Smart Cities. *Transportation engineering*, [online] 16, p.100252. Available at: <https://www.sciencedirect.com/science/article/pii/S2666691X24000277>.

Ferowich, G. (2023). *How to Implement Query Caching with Redis, Express, Node.js, and React*. [online] Medium. Available at: <https://medium.com/@gferowich/how-to-implementquery-caching-with-redis-express-node-js-and-react-3a4c4b530c6>.

Hörcher, D. and Tirachini, A. (2021). A review of public transport economics. *Economics of Transportation*, [online] 25, p.100196. Available at: <https://www.sciencedirect.com/science/article/pii/S2212012221000010>.

Juan, A.A., Corlu, C.G., Tordecilla, R.D., de la Torre, R., and Ferrer, A. (2023). Optimization of Vehicular Networks in Smart Cities: From Agile Optimization to Learnheuristics and Simheuristics. *Sensors*, 23(1), 499. <https://doi.org/10.3390/s23010499>

Lv, X. and Li, M. (2021). Application and Research of the Intelligent Management System Based on Internet of Things Technology in the Era of Big Data. *Mobile Information Systems*, [online] 2021, pp.1–6. Available at: <https://onlinelibrary.wiley.com/doi/pdf/10.1155/2021/6515792>.

Macedo, E., Teixeira, J., Sampaio, C., Silva, N., Coelho, M.C., Glinos, M. and Bandeira, J.M. (2021). Real-time information systems for public transport: user perspective. *Transportation Research Procedia*, [online] 52, pp.732–739. Available at: <https://www.sciencedirect.com/science/article/pii/S2352146521001344/pdf?md5=bda4641aff33eafbaeb9fdf06bbc4b94&pid=1-s2.0-S2352146521001344-main.pdf>.

Mohammadzadeh, H., Dami, S., Shokr, A., Farokhi, V., and Navimipour, N.J. (2024). Digital Transformation for Sustainable Transportation: Leveraging Industry 4.0 Technologies to Optimize Efficiency and Reduce Emissions. *Future Transportation*, 5(2), 34. <https://doi.org/10.3390/futuretransp5020034>

- Oladimeji, D., Gupta, K., Kose, N.A., Gundogan, K., Ge, L., and Liang, F. (2023). Smart Transportation: An Overview of Technologies and Applications. *Sensors*, 23(8), 3880. <https://doi.org/10.3390/s23083880>
- Soumahoro, I., Ceschia, A., Azib, T., Bethoux, O., and Alves, F. (2023). Real-Time Integrated Energy Management Strategy Applied to Fuel Cell Hybrid Systems. *Energies*, 16(6), 2645. <https://doi.org/10.3390/en16062645>
- Subramanian, M., and Tamilselvan, K. (2024). Intelligent Transportation System Technologies, Challenges and Security. *Applied Sciences*, 14(11), 4646. <https://doi.org/10.3390/app14114646>
- Sun, B., and Li, J. (2017). A NoSQL–SQL Hybrid Organization and Management Approach for Real-Time Geospatial Data: A Case Study of Public Security Video Surveillance. *ISPRS International Journal of Geo-Information*, 6(1), 21. <https://doi.org/10.3390/ijgi6010021>
- Taha, A.M., Aldhaferi, S., Musaed, A.A., Alkahtani, A.A., Aljumah, A.I., Alsubai, S., and Saleh, H. (2024). A Roadmap to Systematic Review: Evaluating the Role of Data Networks and Application Programming Interfaces in Enhancing Operational Efficiency in Small and Medium Enterprises. *Sustainability*, 16(23), 10192. <https://doi.org/10.3390/su162310192>
- Vangelista, L., Zanella, A., and Zorzi, M. (2019). A Review of Machine Learning and IoT in Smart Transportation. *Future Internet*, 11(4), 94. <https://doi.org/10.3390/fi11040094>
- Wang, X., Liu, S., Shi, H., Xiang, H., Zhang, Y., He, G., and Wang, H. (2022). Impact of Penetrations of Connected and Automated Vehicles on Lane Utilization Ratio. *Sustainability*, 14(1), 474. <https://doi.org/10.3390/su14010474>