19CSE445 – Cloud Computing

Open Source Blockchain & Cloud Integration for Internet of Vehicles: A Multi-Domain IoT Communication Solution

Project Report

Submitted by

BL.EN.U4EAC22026	Krishna P. Tambatkar
BL.EN.U4EAC22033	Lasyapriya Bharadwaj K.
BL.EN.U4EAC22047	Shashank R.

Submitted to

Dr. Bhavana V.

Assistant Professor (Sl. Gr.)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMPUTER ENGINEERING

AMRITA SCHOOL OF ENGINEERING

AMRITA VISHWA VIDYAPEETHAM

BENGALURU-560035

Abstract

Safety in the Internet of Vehicles (IoV) depends on rapid, trustworthy exchange of nearby hazard information. Prior blockchain-IoV efforts prove immutable logging but often rely on centralized gateways or a single global contract, overlook zone awareness, and expose users to latency and irrelevant or misleading alerts. This project presents AlertNet, a decentralized, zone-aware safety-alert system that couples cloud geofencing with per-zone smart contracts to deliver tamper-evident, locality-specific alerts without a single point of failure. AlertNet follows a three-layer model. The perception layer is a cross-platform React-Native app (Google Sign-In, GPS capture, live map) that shows the current zone and supports structured alert submission. The network layer (AWS API Gateway) exposes two HTTPS endpoints — '/findZone' and '/sendAlert' backed by Lambda functions that query a PostgreSQL (RDS) geofence catalog and orchestrate blockchain publication. The application layer hosts per-zone Ethereum-compatible contracts (Sepolia) that record alerts via gas-efficient event logs; latitude/longitude are integer-scaled for Solidity precision. On device, a zone context applies the Haversine distance to bind the user to the nearest zone, listens for that contract's 'AlertPosted' events in real time, and switches contracts seamlessly as the driver moves. A prototype across seven urban zones achieved seconds-scale end-to-end latency (app $\rightarrow$ cloud $\rightarrow$ chain $\rightarrow$ app), immediate on-chain provenance, and robust operation under intermittent connectivity. Event-only ledgering minimized on-chain cost while preserving auditability; the per-zone design reduced spam and eliminated single points of failure. The open problem of fake alerts and outline pragmatic safeguards—sensor corroboration (camera/LiDAR cues) and lightweight reputation/stake—to flag or down-rank dubious reports. Overall, AlertNet offers a scalable, locality-aware, and tamper-evident substrate for IoV safety messaging suitable for staged public-sector adoption.

Contents

List of Abbreviations

AI – Artificial Intelligence

API – Application Programming Interface

AWS – Amazon Web Services

CPU – Central Processing Unit

DB – Database

GPS – Global Positioning System

HTTP – Hypertext Transfer Protocol

IoT – Internet of Things

IoV – Internet of Vehicles

JSON – JavaScript Object Notation

LIDAR – Light Detection and Ranging

ML – Machine Learning

PUF – Physical Unclonable Function

RDS – Relational Database Service

SQL – Structured Query Language

UI – User Interface

V2V – Vehicle-to-Vehicle

V2X – Vehicle-to-Everything

VANET – Vehicular Ad-Hoc Network

VPC – Virtual Private Cloud

PBFT – Practical Byzantine Fault Tolerance

RL – Reinforcement Learning

ETH – Ethereum (Cryptocurrency / Network)

Gwei – Gigawei (Ethereum Gas Unit)

SSH – Secure Shell

HTTP(S) – Hypertext Transfer Protocol (Secure)

EC2 – Elastic Compute Cloud

GPS – Global Positioning System

API – Application Programming Interface

RSA – Rivest–Shamir–Adleman (cryptography)

NTRU – Nth-degree Truncated Polynomial Ring Unit (lattice-based crypto)

LWE – Learning With Errors (cryptography)

Chapter 1

Introduction

The timely and reliable distribution of hazard information among vehicles, roadside units, and authorities is essential to road safety in the Internet of Vehicles (IoV). A recent survey mapped the range of blockchain-enabled IoV applications, including payments, sensing, energy trading, and traffic management, while highlighting unresolved privacy, trust, and scalability issues [1]. Additional early work showed that blockchain can authenticate V2V messages and support trusted vehicular services without a single point of failure [2]. Adaptive trust management and reputation-led ledgers, which down-weight malicious reporters and stabilize decision-making under attack, were developed by researchers to combat misinformation and adversarial behaviors [3, 4]. Consortium chains and customized consensus techniques preserve tamper-evident traffic data pools and lower confirmation latency in dynamic networks for availability and integrity [5]. Smart-contract provenance, dynamic pseudonyms, and signcryption have all been used to achieve privacy-preserving transactions that allow for responsible data exchange while preventing identity leakage [6, 7]. Future-proofing vehicular communication against device compromise and post-quantum threats has been investigated through the use of hardware-assisted authentication and quantum-safe primitives, further streng-thening the security baseline [8, 9, 10]. Sidechains for bulk storage and reputation-based PBFT address scalability and data management, enhancing network efficiency and throughput in high traffic [11]. At the same time, federated learning and blockchain have been combined to enhance prediction and resource orchestration without centralizing sensitive data, bringing connected mobility performance and privacy requirements into line [12].

Yet a gap persists, most systems secure generic V2X exchanges or long-horizon analytics, while

safety-critical, near–real-time alerting (e.g., drunk/drowsy driving or road obstruction) requires a pipeline that can verify origin, prevent tampering, and deliver **zone-aware** notifications within seconds without relying on a central coordinator. If trust, locality, or latency are compromised, alerts are ignored or, worse, mislead drivers. This motivates a design that jointly optimizes integrity, privacy, and responsiveness for incident reporting and dissemination.

This project presents a cloud-backed, consortium-blockchain safety-alert system for IoV that:

- Partitions the road network into geographic zones to localize dissemination and reduce network load;
- Realizes an end-to-end trusted pipeline structured mobile reporting (vehicle metadata, category, geotag) $\rightarrow$ smart-contract validation (deduplication, reporter eligibility, zone checks) $\rightarrow$ immutable ledgering for auditability; and
- Offers a deployable prototype with live mapping and geolocation, enabling prompt, verifiable, and resilient hazard broadcasts to vehicles subscribed within affected zones.

The further sections discuss security properties, latency characteristics, and practical deployment aspects on a cloud substrate, positioning the system as a focused step toward trustworthy, real-time safety services in IoV.

Chapter 2

Literature Survey

Recent research in blockchain technology has significantly enhanced the efficiency and security of the Internet of Vehicles (IoV). Foundational work by Jabbar et al. (2020) proposed zone-scoped smart contracts combined with cloud integration to enable decentralized real-time alerting. DISV, an Ethereum-based system, furthered this by authenticating vehicle communications and broadcasting real-time alerts through smart contracts [13].

A comprehensive survey by Hildebrand et al. (2023) categorizes numerous blockchain-enabled IoV applications spanning energy trading, crowdsourced sensing, infotainment, and congestion management. Despite these advances, the survey highlights persistent security, privacy, and trust challenges, advocating for emerging technologies like federated learning in vehicular networks [1].

Addressing trust, Shamili et al. (2025) introduced Blockchain-MLTrustNet, combining adaptive graph-sharded blockchains with reinforcement learning, improving trust-score accuracy by 15% and reducing transaction latency by 37% [3]. Juárez and Bordel (2023) developed a dual-blockchain architecture using a Bayesian reputation model that achieved an 86% success rate in mitigating malicious vehicle behavior within VANETs [4].

Security and consensus innovations include Sehar et al. (2023) designing a consortium chain with Vehicular-Based Consensus Algorithm (VBCA) for tamper-proof messaging and enhanced confirmation latency, and Zhang et al. (2024) introducing a V2V electricity-trading scheme leveraging certificateless aggregate signcryption on a reputation-based consortium blockchain for privacy preservation and minimal overhead [5, 6].

Integrity and accountability features are improved by Yin (2022) through a blockchain data-provenance layer enforcing access control policies cost-effectively [7]. Enhancements in authentication are demonstrated by Sahu and Chandrakar (2025) integrating hardware PUFs and Bloom filters to reduce EV-device-to-vehicle authentication delay by 74% and storage by 40% [8]. Quantum-resilient protocols, like Singh et al. (2025) proposing a ring-LWE aggregate-signature, and Yadav et al. (2025) suggesting an NTRU lattice-based multi-signature consensus, address future-proofing security, reducing data size, and improving throughput [10, 9].

Finally, Devarajan et al. (2025) showcase the promising integration of federated learning with blockchain, enabling distributed traffic model training by vehicles secured by blockchain-verified updates, thus enhancing traffic prediction accuracy, integrity, and lowering latency [12].

This body of work collectively illuminates the growing synergy between blockchain, AI, and vehicular networks, underscoring the multidisciplinary approaches essential for advancing secure, scalable, and trustworthy IoV systems.

2.1 Summary

The reviewed literature can be grouped into four methodological clusters for effective synthesis as shown in Table:

Table 2.1: Summary of Methodologies, Contributions, and Limitations

Group	**Papers**	**Contribution**	**Approach**	**Limitations**
Foundational Architectures	[13]	Decentralized IoV alerting with zone-scoped contracts and cloud integration	Ethereum smart contracts, cloud computing	Early frameworks; less focus on scalability and trust models

Continued on next page

Table 2.1 – continued from previous page

Group	Papers	Contribution	Approach	Limitations
Application Survey	[1]	Categorized blockchain IoV applications: energy trading, sensing, infotainment, congestion	Literature survey, federated learning recommendation	Security and privacy challenges remain unresolved
Trust Management	[3, 4]	Enhanced trust scores, reputation models, reduced transaction latency	Adaptive blockchain sharding, reinforcement learning, Bayesian model	Limited in integrating heterogeneous IoV data sources
Security & Consensus	[5, 6]	Tamper-proof messaging, improved latency, privacy-preserving communications	Consortium chains, VBCA, certificateless signcryption	Complexity of key management and consortium governance
Integrity & Accountability	[7, 8]	Access control enforcement, lightweight authentication enhancements	Blockchain data provenance, hardware PUFs, Bloom filters	Potential overhead in hardware integration; scalability concerns
Quantum-Resilient Protocols	[10, 9]	Quantum-secure signatures, reduced verification data size, improved throughput	Ring-LWE, NTRU lattice, PBFT, sidechain	Early-stage prototypes; computational complexity
AI and Federated Learning	[12]	Blockchain-secured federated learning for improved traffic prediction and integrity	Federated learning, blockchain to secure model updates	Integration challenges and real-time data processing

The reviewed literature shows blockchain greatly improves Internet of Vehicles by securing real-time communication, managing trust, preserving privacy, and enabling decentralized, transparent data sharing. It highlights advances in trust models, consensus algorithms, quantum security, and federated learning integration. Despite progress, challenges like real-time data integration, scalability, and off-chain trust remain, guiding future research toward more robust and efficient IoV blockchain applications. This project will build on these insights to develop a scalable, secure, zone-aware alerting platform using blockchain and cloud integration.

2.1.1 Key Takeaways

Key benefits of integrating blockchain technology with the Internet of Vehicles include enhanced data security, privacy, and trust through decentralization and immutability; improved real-time communication among vehicles; elimination of single points of failure; and support for innovative applications like real-time safety alerts and traffic management. Blockchain also facilitates transparent and tamper-proof data sharing, reduces reliance on centralized authorities, and enables secure vehicle authentication. This project leverages these strengths by developing a decentralized, zone-aware alerting system that combines blockchain's trustworthiness with cloud scalability to improve vehicular safety and communication.

2.1.2 Open Challenges

Key open challenges in blockchain-enabled Internet of Vehicles include ensuring privacy protection for sensitive vehicular data, achieving interoperability among diverse IoV devices and platforms, managing high transaction throughput and scalability under dynamic vehicle mobility, reducing latency for timely alert delivery, and addressing the computational complexity of consensus protocols. This project tackles some of these by leveraging zone-specific smart contracts for scalable, efficient alert management and combining blockchain with cloud infrastructure for real-time, secure, and decentralized vehicular communication.

Chapter 3

System Design

This chapter presents a comprehensive exploration of the technical foundation and architecture of the AlertNet system. It methodically examines the rationale behind core technology choices, including AWS Cloud, blockchain fundamentals, and the Sepolia Testnet environment. Following this foundation, the chapter details the multi-layered IoT architecture, describing the perception, network, and application layers, along with the blockchain integration. Each subsection unpacks the component responsibilities, interconnectivity, and workflow dynamics that collectively enable trusted, real-time safety alerting. The chapter culminates in a detailed description of AWS infrastructure configuration and mobile application functionality, drawing a cohesive technical landscape for this decentralized IoT platform.

Before delving into the specific components and implementation details of AlertNet, it is important to first clarify some foundational concepts underlying the overall design. A clear understanding of the core elements—such as AWS Cloud, Blockchain technology, and Sepolia Testnet—sets the stage for exploring how each layer of the system contributes to a scalable, secure, and trustworthy platform for real-time zone-based road safety alerts. These components not only provide the infrastructure backbone but also underpin the integrity and reliability expectations critical to an application designed to protect drivers on the road.

3.1 Background

3.1.1 AWS Cloud

Amazon Web Services (AWS) is a comprehensive and widely adopted cloud computing platform developed by Amazon. Launched in 2006, AWS offers a broad suite of over 200 fully featured cloud services, including computing power, storage options, databases, networking, machine learning, analytics, and security. These services are delivered from a global network of data centers strategically located across geographic regions and availability zones, enabling businesses to deploy applications and services with low latency, high availability, and robust disaster recovery capabilities.

AWS operates on a pay-as-you-go model, allowing organizations—from startups to large enterprises to scale resources dynamically based on their needs, without the overhead of managing physical infrastructure. Core services such as Virtual Private Cloud (VPC) enable secure network segmentation, Amazon RDS provides managed relational databases with automated backups and scaling, Lambda offers serverless compute functions that run on demand, and API Gateway manages secure and scalable API endpoints. This cloud-native architecture reduces operational complexity, enhances security through granular access controls, and supports rapid development and deployment cycles—key advantages for real-time Internet of Things (IoT) systems like AlertNet.

The adoption of AWS within AlertNet ensures that the system can handle fluctuating workloads efficiently while maintaining the security and isolation necessary for sensitive vehicular and location data, ultimately contributing to reliable, scalable, and secure road safety alert management.

3.1.2 Blockchain

The integration of blockchain technology introduces decentralization and immutability to the safety alert records within AlertNet. Blockchain operates as a distributed ledger, ensuring that all transactions—such as posting alerts—are permanently recorded across a network of nodes rather than residing in a centralized database. This decentralization significantly enhances security by reducing vulnerabilities associated with single points of failure and making tampering practically impossible without consensus approval. The immutability offered by blockchain guarantees that once alerts are logged, they cannot be altered or deleted, ensuring a trustworthy

and transparent record of road safety events.

Smart contracts, which are self-executing pieces of code deployed on the blockchain, play a critical role in automating and enforcing alert management. They facilitate the verification, storage, and broadcasting of alerts without intermediaries, minimizing the risk of data manipulation and ensuring timely dissemination. This automated governance caters directly to the trust requirements of vehicular safety applications where the accuracy and integrity of alerts have real-world consequences.

The use of the Sepolia Testnet, an Ethereum-compatible public blockchain test environment, allows AlertNet to develop and validate these smart contracts effectively. Sepolia provides an accessible platform for realistic transaction testing with minimal fees and reasonable confirmation times. Through this setup, AlertNet ensures the reliability and auditability of its alert data, fostering confidence among users and stakeholders.

3.2 System Overview

The AlertNet architecture integrates three synergistic layers to deliver resilient, low-latency, and trustworthy safety alerts for IoV environments. At the perception layer, users interact via a mobile app that continuously acquires geolocation, handles secure sign-in, and presents zone-based hazard data on a live map. The network layer, built on AWS API Gateway, provides a robust middleware that connects mobile clients to the application backend, abstracting authentication, traffic bursts, and network variability. Serverless AWS Lambda functions orchestrate geofence lookups, process alert submissions, and mediate between the PostgreSQL RDS and blockchain domains. At the application layer, per-zone Ethereum smart contracts record events in a distributed ledger, ensuring tamper evidence and verifiable provenance for every safety alert. Figure 3.1 depicts how the mobile perception layer, AWS network layer, and blockchain application layer interconnect to ensure end-to-end alert reliability.

In this architecture, the mobile application captures user context and location, communicating asynchronously over secure APIs with AWS Lambda functions that coordinate database operations and blockchain transactions. Cloud services ensure scalability, while blockchain smart contracts ensure trust and transparency. This flow accommodates real-time alerts with minimal latency. In depth exploration on these layers will be done in the upcoming sections.

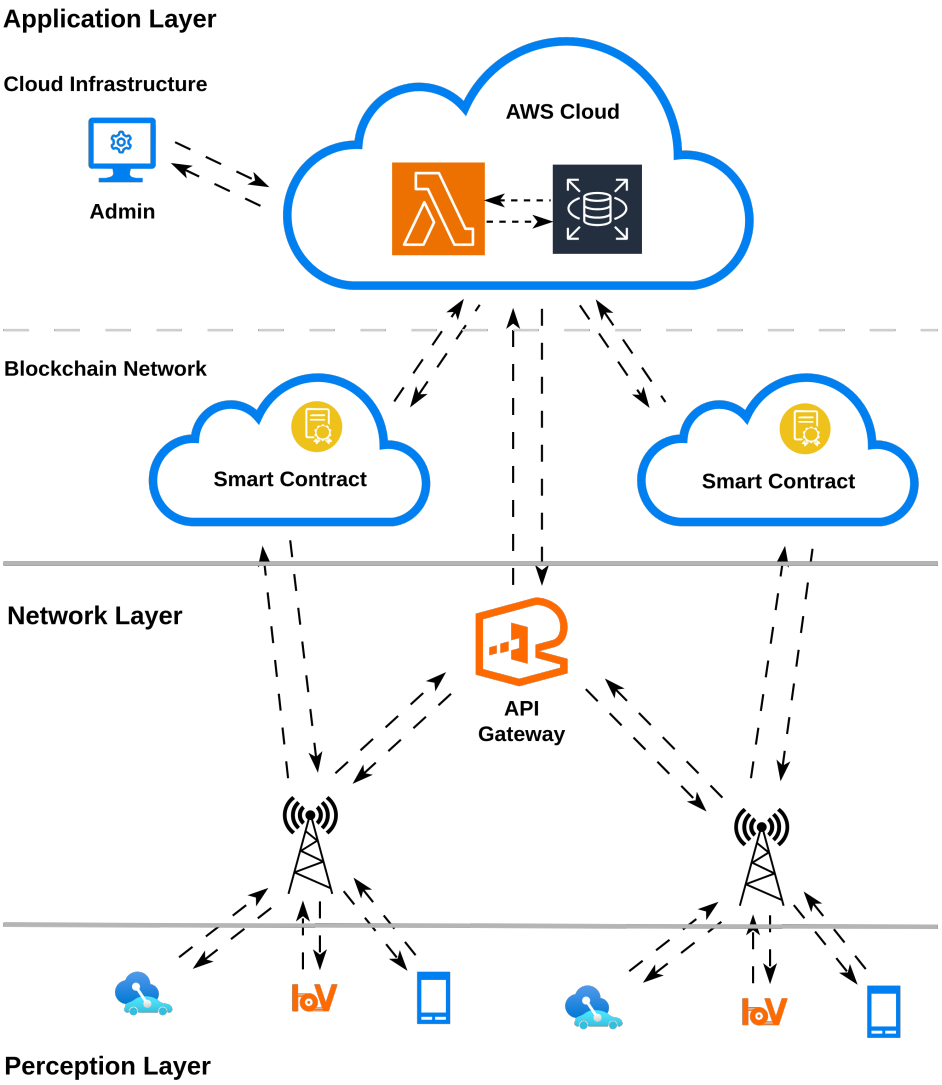

Figure 3.1: AlertNet Layered Architecture.

3.3 The Perception Layer

At the interface with the user, the perception layer delivers critical functionality through the AlertNet mobile application.

3.3.1 Application Description

Built using React Native, the AlertNet app provides cross-platform compatibility for Android and iOS. The app includes Google OAuth 2.0 integration for authentication, guaranteeing secure access to alerting functions.

3.3.2 User Interface and Core Functionality

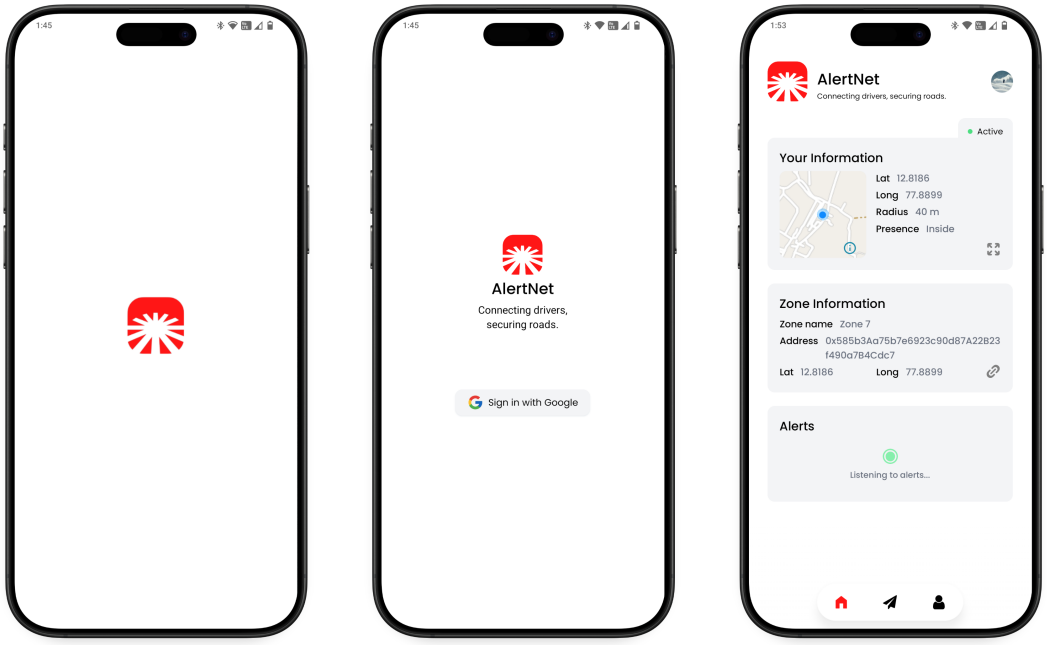

Figure 3.2: Mobile Application Home Screen.

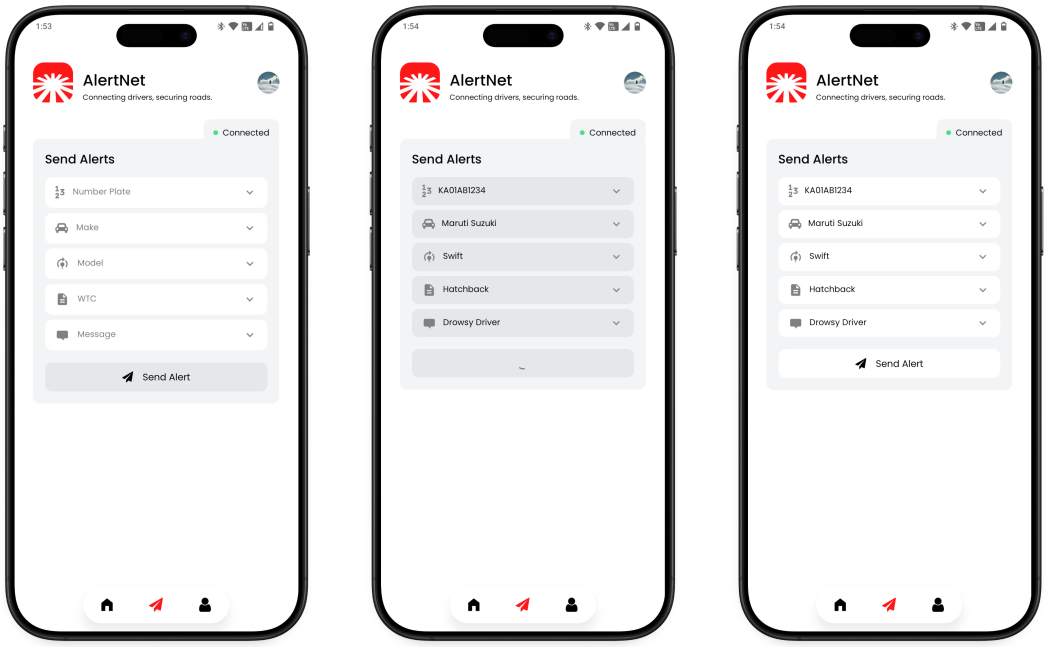

Figure 3.3: Mobile Application Send Alert Screens.

Users are presented with an intuitive home screen displaying current GPS-based zone coordinates, geofence radius, zone presence status, and active incident alerts. Incident reporting is streamlined via user-friendly forms enabling the quick submission of detailed hazard information. Figure 3.2

and 3.3 showcases the user interface presenting geofence data and the hazard alert submission flow.

3.3.3 Location Data and Geofence Checking

The app continuously polls GPS data every 5 seconds or 10 meters using the Expo Location API. It locally computes zone inclusion using the geographically accurate Haversine formula, reducing network overhead and providing timely zone context to the user interface.

3.4 The Network Layer

This layer provides the communication backbone connecting mobile clients with cloud services securely and efficiently.

3.4.1 Secure API Communication

AWS API Gateway exposes two main RESTful endpoints—/findZone and /sendAlert—that accept encrypted JSON inputs. Gateway enforcement ensures authenticated, rate-limited access, safeguarding system integrity.

3.4.2 Lambda Functions

Serverless AWS Lambda functions handle incoming API requests. The findZone function queries the RDS zone database with coordinates, calculating geofence matches. The sendAlert Lambda composes, signs, and submits alert transactions to the blockchain.

3.4.3 VPC Network Setup

The infrastructure operates within an AWS VPC (alertnet-vpc), segmented into two public subnets and two private subnets distributed across two availability zones (ap-south-1a, ap-south-1c). Public subnets host the Bastion Host for secure DB administration, while private subnets house the RDS instance (with strict access control) and Lambda functions (with outbound-only permissions). Figure 3.4 visual representation of subnets, security groups, Lambda functions, Bastion Host, and RDS deployment within the VPC.

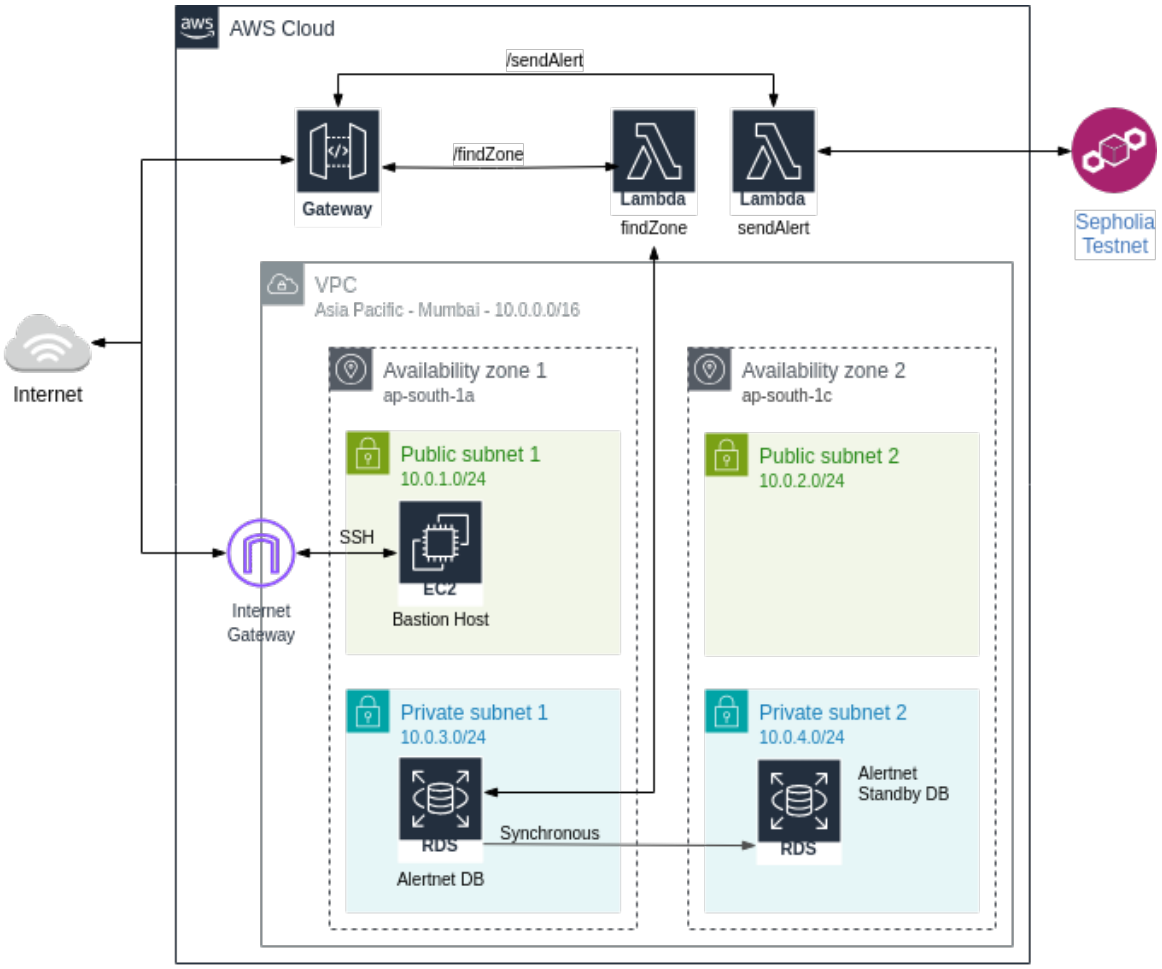

Figure 3.4: AWS VPC and Service Topology.

3.5 The Application Layer

The application layer orchestrates the core processing, data management, and blockchain interactions underpinning AlertNet.

3.5.1 Database Implementation

The PostgreSQL RDS instance (alertnetdb) securely stores zone configuration data, including IDs, names, geographical coordinates, and smart contract addresses. Database access is restricted to Lambda functions and the Bastion Host, conforming to least-privilege principles.

3.5.2 Lambda Functionality

The two AWS Lambda functions fulfill distinct but complementary roles within AlertNet. The findZone Lambda function is responsible for efficiently processing client requests to return

zone metadata based on the user's current location. It performs geospatial queries against the PostgreSQL database hosted on Amazon RDS, utilizing the Haversine formula to calculate the distance between the user's coordinates and stored zone centers, thereby determining which geofence zone the user resides in. This function enables real-time zone detection while optimizing resource usage through precise querying and caching strategies.

The sendAlert Lambda function handles the complete blockchain alert posting workflow. It securely manages environment-stored cryptographic keys required for signing transactions and interacts with the Sepolia Testnet smart contracts corresponding to specific zones. Upon receiving an alert payload, this function forms a transaction to invoke the smart contract's alert posting function, thereby logging the alert immutably on the blockchain. This serverless approach ensures secure, scalable, and cost-effective handling of blockchain interactions without the need for dedicated servers or manual key management.

Together, these Lambda functions maintain a streamlined and secure bridge between user devices, the cloud database, and decentralized blockchain networks, supporting the low-latency, reliable, and trustworthy operation of AlertNet.

3.6 Blockchain Network

This section provides an in-depth examination of the blockchain network architecture used in AlertNet, with a focus on the mechanisms ensuring alert authenticity, transparency, and precise geographic association. It explains the design and role of smart contracts in enabling decentralized and tamper-resistant alert management, and details how unique geofenced zones are mapped to individual contract addresses for scalable, zone-aware alert posting.

3.6.1 Smart Contract Design

The smart contract layer is a cornerstone of the AlertNet system, enabling decentralized, tamper-proof management of road safety alerts. The Solidity contract ZoneAlert.sol encapsulates all contract logic governing alert publication within individual zones.

This contract is designed to efficiently handle streaming alert data from users by emitting events rather than persistently storing large amounts of data on-chain, a choice that significantly reduces transaction costs and improves scalability.

3.6.2 Geofenced Zone and Smart Contract Mapping

For precise geofencing integration with blockchain, each smart contract is mapped to a unique set of latitude and longitude coordinates along with a specific radius, defining the exact area of coverage for each zone. These parameters are crucial for ensuring that alerts posted to the blockchain are verifiably tied to a specific geographic region.

The mapping of latitude and longitude values is performed with high precision, typically scaling coordinates to integer representations suitable for smart contract use, as floating-point arithmetic is not natively supported in Solidity. This approach ensures seamless compatibility and deterministic behavior during zone checks and event emission.

All zone definitions—including zone names, contract addresses, geographic centers, and radii—are stored in the zones table within the alertnetdb PostgreSQL database hosted on AWS RDS. Administrative changes to the zone configuration, such as adding a new geofence or updating coordinates, are securely performed using the bastion account (admin privileges), which connects to the database through a protected SSH tunnel on an EC2 instance. This is both a security best practice and a way to maintain operational integrity over critical zone metadata.

```
alertnet_db=> select * from zones;
 id | zone_name |              contract_address              | latitude  | longitude | radius_meters
----+-----------+--------------------------------------------+-----------+-----------+---------------
  2 | Zone 2    | 0x0b1ae92B05eAF806206507502E4aB1aa1270d259 | 12.894232 | 77.674769 |            35
  4 | Zone 4    | 0xF67619a6e0827A08b0111a561A85e7DFf5c1E78E | 12.897731 | 77.676161 |           100
  1 | Zone 1    | 0x433C5b7a9c2982Ec3127330Bb75f75FE736e1CB8 | 12.894766 |  77.67585 |            40
  3 | Zone 3    | 0xdd7AF4aeE31AfF1e7c04616A6399f2284861adaa | 12.894201 | 77.675362 |            30
  6 | Zone 6    | 0x07aD5b46E153fcc92407843b635355840b1138e8 | 12.894188 | 77.675762 |            20
  7 | Zone 7    | 0x585b3Aa75b7e6923c90d87A22B23f490a7B4Cdc7 | 12.894889 | 77.675064 |            40
  5 | Zone 5    | 0x4F05b2e1e0E3e06e4477876604895167DbdfD5D0 | 12.893809 | 77.675083 |            30
(7 rows)

alertnet_db=>
```

Figure 3.5: Contents of the zone table in RDS.

Figure 3.5) shows an example query output from the zones table, highlighting the structured storage of each zone's blockchain contract address and geographic parameters. This architecture enables the AlertNet system to decisively associate any blockchain alert with a physical location, forming the backbone for reliable, transparent, and zone-specific alert delivery.

3.7 System Workflows

This section discusses the core workflows that enable AlertNet to function effectively in real-world driving scenarios. It covers how the system detects a driver's current geofence zone based on GPS updates and cloud queries, the transmission and logging of user-generated alerts via blockchain smart contracts, and the dynamic management of zone transitions as drivers cross geofence boundaries. These workflows ensure users receive timely, context-relevant alerts without interruption, thereby enhancing road safety through precise, real-time notifications.

3.7.1 Zone Detection

The zone detection process begins whenever the device registers a GPS update, either due to significant location change, attained accuracy, or elapsed time (such as every 60 seconds). Upon such an update, the mobile device sends its latitude and longitude coordinates via an HTTP request to the cloud function exposed by the /findZone endpoint, implemented as an AWS Lambda function. The payload sent to the /findZone endpoint is shown in Listing 3.1.

Listing 3.1: Example User Coordinates Sent as Payload

```
{
        "latitude": 12.894232,
        "longitude": 77.674769
}
```

The Lambda function connects securely to the PostgreSQL database hosted on Amazon RDS, which contains the master list of defined zones with their geographic coordinates and radius values. The function executes a geospatial SQL query using the Haversine formula directly in the query as shown in Listing 3.2, to calculate the great-circle distance between the submitted coordinates and each zone center.

Listing 3.2: SQL Query for Nearest Geofence

```
SELECT id, zone_name, contract_address, latitude, longitude,
   radius_meters,
(6371000 * acos(
cos(radians($1)) * cos(radians(latitude)) *
cos(radians(longitude) - radians($2)) +
sin(radians($1)) * sin(radians(latitude))
```

```
)) AS distance_m
FROM zones
ORDER BY distance_m ASC
LIMIT 1;
```

If this minimum distance is less than the zone's radius, the Lambda function returns zone details, including the blockchain smart contract address associated with that location. The mobile app caches this zone data locally to reduce redundant requests, falling back to querying only when relevant location changes occur or timeout thresholds are reached.

Listing 3.3: Example Response from /findZone Endpoint

```
{
        "zoneName": "Zone_2",
        "contractAddress": "0x0b1ae92B05eAF806206507502E4aB1aa1270d259"
           ,
        "latitude": 12.894232,
        "longitude": 77.674769,
        "radius": 35
}
```

An example response is shown in Listing 3.3. Upon successfully retrieving and confirming the current zone, the device subscribes to the blockchain smart contract at the provided contract address, listening for real-time alerts scoped to that geofence. This mechanism assures that the driver continuously receives contextually relevant notifications tailored to their specific location without lag or unnecessary network overhead.

3.7.2 Alert Broadcasting

User-generated alerts are transmitted to the blockchain network where smart contracts serve as decentralized, automated systems that log these alerts as events. The AlertNet mobile application actively listens to these smart contract events using blockchain event subscription mechanisms. Whenever a new alert message is emitted by the contract, the mobile app detects it in real time and generates a local notification, immediately alerting the user to the new safety concern.

Beyond passive notifications, the alert is also represented visually within the app. Newly

received alerts populate the alert feed on the Home Screen, providing users with a history and overview of recent incidents. Additionally, the geographic location of the alert is mapped on an expanded map view embedded in the Home Screen, allowing users to see exactly where the event occurred relative to their current position or zone.

This integration ensures that users receive timely, actionable updates while also enabling a spatial understanding of hazards around them, strengthening situational awareness and facilitating safer driving behavior. Figure 3.6 illustrates app UI reacting to live incident notifications in active zones.

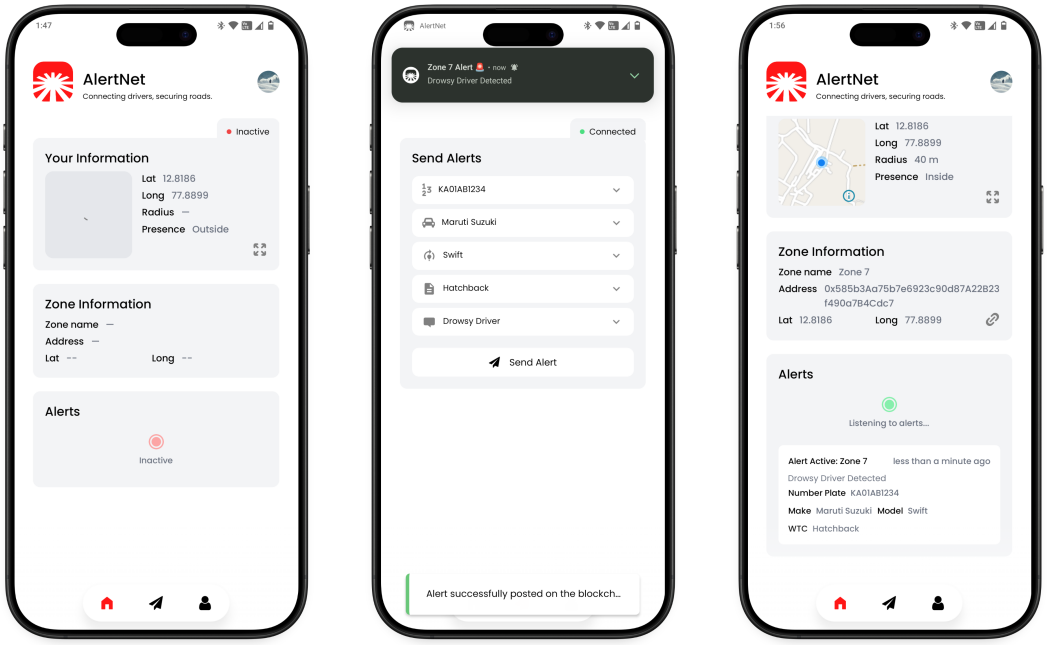

Figure 3.6: Mobile App Alert Notification Screens.

3.7.3 Zone Transition Management

The system dynamically detects whether a driver crosses geofence boundaries, ensuring that alerts are delivered only when relevant. This is achieved using geofencing, which defines virtual geographic boundaries around specific zones. When a driver's GPS location crosses these boundaries, the system triggers context-aware operations like switching alert subscriptions to the current zone.

Mathematically, the geofence is modeled as a circle defined by a **center point** with coordinates (ϕ_c, λ_c) (latitude and longitude) and a **radius** r. The driver's position is given by (ϕ, λ).

To determine whether the driver is inside or outside the geofence, the system computes the

great-circle distance d between the driver's location and the geofence center using the **Haversine formula** shown as Equation 3.1, which accounts for Earth's curvature:

$$d = 2r_E \arcsin\left(\sqrt{\sin^2\left(\frac{\phi - \phi_c}{2}\right) + \cos(\phi_c)\cos(\phi)\sin^2\left(\frac{\lambda - \lambda_c}{2}\right)}\right) \tag{3.1}$$

Where:

- d is the distance between two points on the Earth's surface,
- r_E is the Earth's radius (approximately 6,371 kilometers),
- ϕ_c, ϕ denote the latitudes in radians,
- λ_c, λ denote the longitudes in radians.

If the calculated distance d satisfies the condition:

$$d \leq r \tag{3.2}$$

then the driver is considered inside the geofence. This calculation enables precise and efficient boundary crossing detection, triggering zone transition workflows essential for delivering timely and relevant safety alerts.

This chapter has detailed the end-to-end design of AlertNet, highlighting how its layered architecture and decentralized principles create a robust, secure, and scalable platform for real-time safety alerting in the Internet of Vehicles. By coupling an intelligent mobile interface with secure cloud APIs and per-zone blockchain smart contracts, the system achieves precise geofencing, low-latency notification delivery, and tamper-evident auditability. The synergistic integration of cloud and blockchain infrastructure ensures resilience, transparency, and seamless user experience even under dynamic network conditions.

The next chapter, Results and Discussion, will evaluate AlertNet's effectiveness through measured metrics—such as alert latency, transaction costs, and reliability while also discussing encountered limitations and avenues for future enhancements in decentralized vehicular safety systems.

Chapter 4

Results and Discussion

4.0.1 Latency and Confirmation Time

Latency in the AlertNet system is defined as the time elapsed between the submission of an alert transaction to the blockchain and its successful confirmation—that is, the alert's inclusion in a mined block. This latency directly affects how quickly users receive verified safety notifications. Measurement of **50 alert transactions** showed an **average confirmation time of 9.4 seconds**, with the **shortest confirmation at 6.4 seconds** and the **longest at 11.5 seconds**. This efficient confirmation time ensures that alerts are delivered promptly, minimizing user wait time and maximizing timely awareness of hazards.

The confirmation time is influenced by blockchain factors including network congestion, miner behavior, and transaction fee settings. Despite these variables, the observed latencies remain well within reasonable limits for real-time IoT alerting applications, demonstrating AlertNet's suitability for live safety-critical deployments.

4.0.2 Average Gas Cost

The Ethereum gas fees for posting alerts on the Sepolia testnet are relatively low, averaging around **0.00000003 ETH** per transaction. This minimal fee ensures that each alert can be posted cost-effectively, supporting the system's scalability and operational sustainability. The current gas prices on Sepolia are roughly **0.1 Gwei**, which translates to a very small transaction cost—generally well within the limits of typical test scenarios. Such low costs make it feasible to deploy and maintain a large-scale alerting system without significant expense, promoting

experimentation and reliable operation during the development phase.

4.0.3 Success Rate of Alert Posting

The reliability of alert submissions is a critical metric for the AlertNet system. Testing revealed that **96%** of alert transactions submitted to the Ethereum Sepolia Testnet were successfully mined, stored, and made accessible on the blockchain. Failed attempts were rare and typically associated with temporary network issues or "Service Unavailable" errors, reflecting external infrastructure limitations rather than system design flaws. This high success rate demonstrates the robustness and dependability of the blockchain alert posting pipeline, ensuring that most safety alerts are reliably recorded and available for user notifications.

4.0.4 Limitations and Future Work

This section identifies current system constraints, such as vulnerability to false alerts, and proposes ongoing and future enhancements. Suggested improvements include integrating deep learning models with vehicle sensor data for fake alert validation, leveraging real-time onboard sensor monitoring for continuous verification, and implementing verification mechanisms in the app to promote accountability and prevent fraud.

While AlertNet demonstrates strong performance in decentralized alert management and real-time location-aware notifications, several limitations remain. The system currently depends primarily on user-generated alerts, making it susceptible to false reports and potential misuse. To address this, future work will focus on integrating deep learning models that utilize vehicle camera and LIDAR data to verify driving behaviors indicative of hazardous conditions, such as aggressive lane switching. Continuous real-time monitoring via onboard sensors will enhance the system's ability to correlate sensor data with reported alerts, ensuring authenticity and reducing false positives. Additionally, incorporating verification mechanisms directly into the mobile app will promote accountability, prevent fraudulent alerting, and support fairness, which is essential for user trust and governmental adoption. These enhancements aim to improve system reliability, safety, and acceptance in increasingly complex vehicular environments.

Chapter 5

Conclusion

The AlertNet system successfully combines blockchain technology with cloud computing to provide a secure, transparent, and decentralized vehicle safety alert network. It ensures tamper-proof broadcasting of real-time, zone-based notifications that enhance driver awareness and safety within the Internet of Vehicles framework. The integration of scalable AWS cloud services with immutable blockchain smart contracts eliminates centralized points of failure, enabling verifiable and traceable communication between vehicles. This modular architecture lays a robust foundation for future expansions, including AI-based alert verification and widespread government adoption, positioning AlertNet as a pioneering model for next-generation road safety solutions.

References

[1] B. Hildebrand et al. “A comprehensive review on blockchains for Internet of Vehicles: Challenges and directions”. In: *Computer Science Review* 48 (2023).

[2] R. Radhakrishnan et al. “Blockchain for the Internet of Vehicles”. In: *Advances in Computing and Data Sciences (ICACDS)* 1045 (2019).

[3] F. S. Shamili and R. Gopi. “Optimizing IoV cloud trust with adaptive blockchain and edge intelligence”. In: *Scientific Reports* 15 (2025).

[4] R. Juárez and B. Bordel. “Augmenting VANET Security and Efficiency with Blockchain: A Probabilistic Identification and Malicious Node Mitigation Strategy”. In: *Electronics* 12 (2023).

[5] N. U. Sehar et al. “Blockchain enabled data security in vehicular networks”. In: *Scientific Reports* 13 (2023).

[6] S. Zhang et al. “A V2V electricity transaction scheme with privacy protection based on the IoV and consortium blockchain”. In: *International Journal of Electrical Power & Energy Systems* 157 (2024).

[7] F. Yin. “A Data Provenance Scheme Based on Blockchain for Internet of Things”. In: *Proc. IEEE Conference on Computer Science and Blockchain (CCSB)*. IEEE. 2022, pp. 42–47.

[8] B. L. Sahu and P. Chandrakar. “Secure blockchain-based IoV architecture integrated with PUFs for secure vehicular communication”. In: *Computers & Electrical Engineering* 123 (2025).

[9] A. S. Yadav et al. “IoT-enabled blockchain framework for Internet of Vehicles safety monitoring in smart cities”. In: *Transactions on Emerging Telecommunications Technologies* 36 (2025).

[10] R. Singh et al. “Building scalable and quantum attack resistant authenticated message delivery system for IoV with blockchain consensus”. In: *Concurrency and Computation: Practice and Experience* (2025).

[11] D. K. Yadav et al. “NTRU lattice–based blockchain consensus–assisted V2V authenticated message delivery in IoV”. In: *International Journal of Communication Systems* (2025).

[12] G. G. Devarajan et al. “Federated learning and blockchain-enabled framework for traffic rerouting and task offloading in IoV”. In: *IEEE Transactions on Consumer Electronics* (2025).

[13] R. Jabbar et al. “Blockchain for the Internet of Vehicles: a decentralized IoT solution for vehicles communication using Ethereum”. In: *Sensors* 20.14 (2020), p. 3928.