JSUCIT

Journal of Shaqra University for Computing and Information Technology
ISSN 2961-4279 (Print) || ISSN 2961-4287 (Online)

Blockchain-based Secure Communication and Coordination Protocol for Electric Vehicle Charging using Drone-assisted Mobile Charging Stations

Someah Alangari[1*]
[1]Computer Science Department, College of Science and Humanities,
Shaqra University, Aldawadmi 11961, Saudi Arabia
Corresponding Author Email[*]: *salangari@su.edu.sa*

Abstract— Drone-assisted mobile charging can extend electric-vehicle (EV) operability in areas where fixed charging infrastructure is sparse or congested, but it introduces new challenges in multi-party coordination, data integrity, and secure settlement. This paper presents a consortium-blockchain protocol for secure communication and coordination in drone-assisted EV charging. EVs and drones operate as lightweight clients, while charging providers and fleet stakeholders operate validator nodes using a low-latency Byzantine-fault-tolerant proof-of-stake style consensus. The protocol defines (i) role-based registration, (ii) signed charging requests and drone status updates, (iii) an assignment workflow with explicit confirmation timeouts, and (iv) an escrow-based payment contract that releases funds only after verifiable charge completion. To reduce on-chain overhead, high-rate telemetry is kept off-chain and anchored on-chain via cryptographic hashes. In simulation (100 EVs in 50 *km*, 10 drones), the proposed approach achieves 8 s average request-to-assignment latency under normal load and 12 s under high load, outperforming centralized and non-blockchain baselines; it also reduces mean travel distance and lowers per-session energy usage (4.0 kWh vs. 4.2 kWh centralized). These results suggest that consortium blockchain can improve auditability and resilience without prohibitive latency for moderate-scale deployments.

Keywords—Blockchain, Communication Protocol, Coordination, Drone-assisted Charging, Electric Vehicle (EV), Mobile Charging Stations, Security

1. INTRODUCTION

The rapid growth of electric vehicles (EVs) is increasing the operational and security requirements placed on charging networks. Even as charging infrastructure expands, practical deployments continue to face uneven spatial coverage, peak-hour congestion, and grid-side constraints that can translate into waiting-time uncertainty and localized overload risk [1, 2]. At the same time, charging infrastructure has become a cyber-physical service ecosystem that must protect identity, metering, and payment workflows against adversarial manipulation [2, 3].

A complementary direction is *mobile* charging, where energy is delivered by roaming ground vehicles or aerial platforms to reduce range anxiety and improve service availability outside dense station deployments. Recent blockchain-enabled designs have explored mobile charging stations (MCS) coordinated through auctions and incentive mechanisms, aiming to improve allocation efficiency and settlement transparency [4, 5]. In parallel, UAV-assisted energy/charging scenarios have been studied in challenging environments (e.g., disaster areas), where intermittent connectivity motivates secure *offline* transaction support and robust accountability mechanisms [6]. UAVs have also been considered as edge-enabled actors in blockchain-supported charging and energy trading decisions (e.g., secure charging-station selection in V2G settings) [7]. These lines of work collectively motivate drone-assisted EV charging as a flexible service layer—but they also underscore the need for secure, low-latency coordination across many parties.

Despite these opportunities, drone-assisted charging amplifies coordination complexity and expands the attack surface. EV charging backends and protocols have documented security weaknesses (e.g., authorization flaws, insecure defaults, and incomplete protection in legacy deployments), which can be exploited to disrupt sessions or manipulate billing [2, 8]. Privacy is also a first-class concern: fine-grained meter readings and charging histories enable profiling unless appropriate privacy-preserving extensions and data-handling practices are used [9]. In drone-assisted settings, additional risks emerge from mobility and real-time decision-making: adversaries can attempt spoofing, false request injection, denial-of-service, or settlement disputes that cannot be reliably adjudicated without trustworthy logs.

Distributed ledger technology (DLT) and smart contracts provide building blocks for tamper-evident logging, multi-stakeholder coordination, and automated settlement. Permissioned/consortium DLT has been used to reduce misbehavior and enforce compliance in decentralized EV-to-station assignment [10], while privacy-preserving charging reservation/payment protocols have been proposed to reduce linkability between blockchain identities and real-world users [11].

Practical charging workflows have also been paired with off-chain storage (e.g., IPFS) to reduce on-chain data overhead while maintaining verifiable integrity anchors [12]. Finally, recent work highlights that improving throughput under Byzantine behavior requires careful consensus and protocol engineering [13], and that high-frequency charging/billing scenarios benefit from scalable designs (e.g., on-/off-chain combinations) [14, 15].

Scope of this work: This paper focuses on the coordination, trust, and settlement layer of drone-assisted EV charging. We assume (i) each EV and drone holds a long-term cryptographic key pair in protected storage, (ii) drones can report coarse-grained location and operational state over wireless links (e.g., 4G/5G/IoT), and (iii) the physical energy-transfer mechanism (cabled DC charging, swappable battery module, or inductive coupling) is abstracted as a charging session characterized by delivered energy and service time. High-rate telemetry is not stored on-chain; it is stored off-chain and anchored on-chain via hashes for auditability.

Research Problem and Objectives: The core problem is to design a secure, auditable, and low-latency coordination framework that can handle drone mobility, real-time EV charging demands, and multiple stakeholders without relying on a single trusted coordinator. Specifically, we aim to:

- Derive security requirements from an explicit system/adversary model (authentication, integrity, non-repudiation, availability, and privacy-by-design).
- Specify a consortium-blockchain architecture suitable for micro-transactions, including validator roles and a principled on-chain/off-chain responsibility split.
- Design smart-contract workflows (registration, request, assignment, escrow/payment, timeout/dispute) that are implementable and resistant to common contract-level risks.
- Design and evaluate a coordination mechanism that assigns drones to EV requests under battery and distance constraints.
- Validate performance and robustness using reproducible simulation scenarios and clearly defined metrics (latency, throughput, energy, utilization, and cost).

Contributions of the Paper:

- A protocol specification for drone-assisted EV charging built on a consortium blockchain, including message flow, role separation, and an integrity anchoring strategy for off-chain telemetry.
- A smart-contract state machine (registration $\rightarrow$ request $\rightarrow$ assignment $\rightarrow$ confirmation $\rightarrow$ charge completion $\rightarrow$ escrow settlement) with explicit timeouts and event logging for auditability.
- A priority-based coordination mechanism for matching drones to EV requests and a discussion of computational feasibility and scalability measures.
- A simulation-based evaluation comparing latency and operational efficiency against centralized and non-blockchain baselines under normal and high-load conditions.

The remainder of this paper is structured as follows: Section 2 reviews EV charging infrastructure, mobile/drone-assisted charging, and prior blockchain-based solutions. Section 3 details the proposed consortium-blockchain protocol and smart-contract workflows. Section 4 analyzes security goals, threat models, and countermeasures. Section 5 presents the coordination mechanism. Section 6 evaluates performance via simulation. Section 7 illustrates a deployment-oriented case study. Section 8 discusses limitations and future work, and Section 9 concludes the paper.

2. BACKGROUND AND RELATED WORK

2.1. Existing EV Charging Infrastructure

Modern EV charging networks comprise interconnected charging points, back-office management systems, user-facing applications, and payment/identity services. As adoption grows, these systems face operational challenges (e.g., demand surges and load balancing) and cybersecurity challenges (e.g., integrity of metering, correctness of authorization, and resilience to network attacks) [1, 2]. Blockchain-based smart-contract approaches have been proposed for charging infrastructures in smart-city settings to reduce single points of failure and enable tamper-evident session/payment records [3].

Beyond fixed stations, *mobile* charging has emerged as an alternative paradigm to improve coverage and reduce waiting times in congestion hot spots. Blockchain-enabled market mechanisms for mobile charging station allocation (e.g., auction-based designs) aim to improve truthfulness, transparency, and settlement automation [4]. Cooperative charging platforms have also been proposed where charging stations share information with verification groups and can deploy mobile stations to reduce waiting time [5]. These approaches help motivate the need for coordination protocols that remain secure under mobility and high request volumes.

Finally, emerging charging modalities (e.g., dynamic charging) raise additional requirements for authentication, privacy, and fair billing while vehicles remain in motion. Smart-contract-based designs have been proposed to support continuous authenticate-and-charge workflows while protecting mobility privacy [16]. Although our work targets drone-assisted mobile charging rather than roadway dynamic charging, both settings share core challenges: fast, repeated interactions, privacy constraints, and a need for automated settlement.

2.2. Role of Drones in Mobile Charging Systems

Drone-assisted charging extends the idea of mobile charging by adding aerial mobility, which can be beneficial in remote regions, traffic-congested urban areas, or emergency

response scenarios. In particular, UAV-assisted schemes have been proposed for secure *offline* charging transactions among EVs in disaster areas, addressing intermittent connectivity and highlighting the importance of accountability and attack resistance [6]. UAVs have also been used as part of edge-enabled decision workflows for blockchain-supported charging and energy trading (e.g., secure charging-station selection in V2G environments) [7]. These studies demonstrate feasibility and motivate a broader protocol that supports real-time assignment, secure communication, and settlement for drone-to-EV charging sessions.

2.3. Communication Protocols for EV Charging

In practice, EV charging ecosystems often rely on standardized communication between chargers and back-office systems. OCPP is widely deployed, and its security posture has been comprehensively analyzed, including threat surfaces and mitigation guidance (with particular attention to legacy deployments and incomplete security hardening) [8]. However, baseline protocol designs typically assume stationary infrastructure and do not natively cover drone-to-EV interaction patterns, mobility-driven reassignment, or multi-operator coordination.

Privacy-preserving extensions to charging communication are also actively studied. For example, private metering mechanisms have been proposed as OCPP extensions to reduce profiling risks from meter readings while maintaining aggregate values needed for operational management [9]. These results reinforce that a drone-assisted charging protocol must treat privacy and secure data handling as core design objectives rather than optional add-ons.

2.4. Security and Scalability in Existing Solutions

EV charging networks face multi-layer security challenges spanning device, network, and application layers, including identity spoofing, false data injection, denial-of-service, malware, and payment fraud [2]. Centralized architectures simplify control but create high-value single points of failure. Conversely, decentralized approaches can improve resilience and auditability but must address latency, throughput, and resource constraints—especially in mobile settings with frequent micro-interactions.

Blockchain-based charging designs increasingly combine smart contracts with scalability techniques. Off-chain storage and integrity anchoring (e.g., IPFS + on-chain hashes) reduce ledger bloat while preserving auditability [12]. For high request volumes, throughput and Byzantine robustness depend on consensus design; recent work proposes more efficient consensus protocols tailored to privacy-preserving charging while maintaining resilience under Byzantine behavior [13]. More broadly, scalable on-/off-chain designs have been proposed to support high-frequency micro-trading scenarios, including EV-related case studies [14], and micropayment-oriented DLT designs have been demonstrated for charging/billing automation [15]. These findings motivate our choice of a consortium setting with lightweight clients and an explicit on-chain/off-chain partition.

2.5. Related Works on Blockchain-Empowered EV Charging

Prior blockchain-enabled EV charging research primarily addresses (i) station discovery/recommendation and assignment, (ii) reservation and payment automation, (iii) privacy preservation, and (iv) load balancing/market coordination. Permissioned DLT has been used to enforce compliance and reduce misbehavior in decentralized charging-station assignment, validated via simulation in realistic mobility settings [10]. Reservation and payment protocols have been designed with stronger privacy guarantees (e.g., ring signatures, SMC-assisted slot verification, and dispute handling with TEEs) [11]. Charging workflow coordination has also been integrated with off-chain storage and smart-contract security testing tools to support practical deployment [12].

In addition, several works explore mobile charging as a service. Auction-based smart-contract frameworks aim to allocate mobile charging stations efficiently and discourage price manipulation [4], while cooperative charging platforms incorporate mobile stations to reduce waiting times and improve information sharing across providers [5]. DLT-backed mobile charging system prototypes further highlight end-to-end settlement and traceability needs [17]. Meanwhile, UAV-assisted charging/energy service studies focus on specialized contexts such as disaster-area V2V charging and UAV-assisted station selection [6, 7].

2.6. Limitations of Current Communication and Coordination Protocols

Despite substantial progress, existing approaches often exhibit one or more limitations when applied to drone-assisted EV charging:

1. **Mobility-aware coordination gaps:** Many protocols assume stationary chargers or ground-based mobile stations, and do not fully address aerial mobility, intermittent connectivity, and rapid reassignment needs.
2. **End-to-end settlement under disputes:** Several designs provide payment automation but do not systematically integrate timeouts, confirmations, and dispute workflows suitable for highly dynamic field operations.
3. **On-chain overhead and privacy:** Naively recording fine-grained telemetry and location traces on-chain increases cost and privacy risk; practical designs require careful on-/off-chain partitioning [9, 12].
4. **Consensus/throughput under adversaries:** High request volumes and Byzantine threats require consensus designs that preserve low latency and robust finality [13, 14].

2.7. Research Gap and Motivation

Overall, the literature supports blockchain as an enabling layer for secure coordination and settlement in EV charging, and it also demonstrates the promise of mobile (and UAV-assisted) charging in specialized scenarios. However, there remains a gap in *integrated* protocol specifications that jointly address (i) drone/EV role separation with lightweight clients, (ii) low-latency consortium validation, (iii) explicit assignment and confirmation state machines, and (iv) escrow-based settlement with verifiable completion and dispute handling. This paper is motivated by that gap and proposes a consortium-blockchain communication and coordination protocol tailored to drone-assisted EV charging.

3. PROPOSED BLOCKCHAIN-BASED PROTOCOL

3.1. Design Goals and System Assumptions

Drone-assisted EV charging requires secure multi-party coordination under mobility, intermittent connectivity, and high-frequency micro-interactions. We design the protocol to achieve: (i) *authentication and non-repudiation* for all critical actions, (ii) *tamper-evident auditability* for assignment and settlement, (iii) *low-latency confirmation* suitable for field operations, (iv) *privacy-by-design* by minimizing on-chain exposure of sensitive telemetry, and (v) *scalability* via a principled on-chain/off-chain partition.

We assume EVs and drones hold long-term public/private keys in protected storage and communicate over wireless networks (e.g., 4G/5G/IoT). The physical energy-transfer mechanism (cabled charging, battery module swap, or inductive coupling) is abstracted as a charging session described by delivered energy and service time. Security requirements and attack surfaces are consistent with those documented for EV charging backends and protocols (e.g., authorization and legacy-hardening gaps in OCPP deployments) [2, 8], and privacy risks stemming from fine-grained metering and charging traces [9].

3.2. System Architecture Overview

The protocol adopts a consortium-blockchain architecture with four roles (Fig. **??**):

1. **Electric Vehicles (EVs)** act as clients that submit signed charging requests and confirm service completion. EVs do not participate in consensus.
2. **Drones (Mobile Charging Units)** act as clients that publish signed availability updates and confirm assignments and completion. Due to resource constraints, drones do not run full blockchain nodes.
3. **Charging Providers / Fleet Stakeholders** operate gateways and validator nodes. Gateways relay client transactions, perform off-chain matching, and store off-chain telemetry. Validators execute consensus and maintain the replicated ledger.
4. **Consortium Blockchain Network** hosts smart contracts for registration, request/assignment, and escrow settlement. Consortium governance distributes trust across stakeholders and avoids single-point-of-failure control, consistent with prior permissioned DLT use in EV assignment problems [10].

3.2.1. On-chain vs. off-chain data

To balance auditability and privacy, we store only *session-critical metadata* on-chain: (i) identities/roles (public keys, permissions), (ii) request/assignment/confirmation events, and (iii) escrow/settlement state transitions. High-rate telemetry (precise trajectories, detailed charging waveforms, sensor logs) is stored off-chain and anchored on-chain via cryptographic hashes and content pointers, following integrity anchoring patterns used in EV synchronization frameworks [12].

3.3. Protocol Transactions and Timing

A charging session is represented by the following signed transactions. Each transaction includes a nonce and timestamp for replay protection.
TxReg: Register or update an identity (EV/drone/provider), role, and public key(s).
TxStatus: Drone status update (coarse location, deliverable energy estimate, state: available/en-route/charging).
TxRequest: EV request containing coarse location, requested energy (or service class), urgency indicator, and maximum waiting time.
TxAssign: Assignment proposal linking a request to a drone, valid for a bounded confirmation window T_{confirm}.
TxAcceptEV / TxAcceptDrone: Mutual acceptance by both parties within T_{confirm}; otherwise the assignment expires.
TxComplete: Completion record containing (a) delivered-energy summary, (b) dual signatures (EV and drone), and (c) hash pointer to the off-chain telemetry bundle.
TxSettle: Escrow release and payment finalization.
TxDispute / TxResolve: Optional dispute workflow when completion reports mismatch beyond a tolerance threshold.

This explicit message taxonomy makes latency definitions unambiguous. In our evaluation, request-to-assignment latency is measured from confirmed **TxRequest** to confirmed **TxAssign** plus both acceptances.

3.4. Smart Contract State Machine

Smart contracts implement a state machine that automates session lifecycle and settlement, while preventing common contract-level risks (unbounded loops, unsafe external calls, ambiguous finalization).

3.4.1. Contract modules

(1) RegistrationContract. Maintains mappings *address* $\rightarrow$ *role* $\rightarrow$ *public key*, supports key rotation/revocation, and enforces onboarding rules (e.g., provider approval and optional security deposit).
(2) RequestContract. Creates and tracks EV requests with bounded lifetimes. It may enforce rate limits and refundable deposits to deter request flooding (Sybil-style abuse).
(3) AssignmentContract. Records drone–EV assignments and enforces mutual confirmation within T_{confirm}. Invariants include: (i) a drone cannot be assigned to two active requests simultaneously, and (ii) expired assignments free the drone for reassignment.
(4) PaymentContract (Escrow). Locks funds upon mutual acceptance and releases them only after verifiable completion. Completion requires dual signatures over session metadata and a hash pointer to off-chain evidence. For stronger privacy and reduced linkability, reservation/payment schemes can incorporate privacy-preserving primitives as in recent blockchain-based EV reservation/payment research [11].

3.4.2. Dispute handling

If EV and drone completion summaries disagree beyond tolerance, the contract transitions to a *DISPUTE* state and allows a provider-operated arbitrator (or consortium committee) to resolve using the off-chain evidence bundle. This preserves auditability without storing sensitive telemetry on-chain.

3.5. Consensus and Network Configuration

We instantiate the protocol on a **consortium blockchain** operated by charging providers and fleet stakeholders. EVs and drones are *lightweight clients*. Validators run a low-latency, BFT-style proof-of-stake / delegated-stake configuration to achieve fast finality appropriate for micro-interactions. Efficient and privacy-aware EV charging protocols emphasize that throughput and Byzantine robustness depend critically on protocol engineering and consensus choices [13]. For scalability, the protocol is compatible with on-/off-chain combinations used in large-scale wireless sharing economies and high-frequency transaction settings [14], and with micropayment-oriented designs applicable to autonomous/electric vehicle settings [15].

3.6. Implementation Considerations

3.6.1. Lightweight clients for drones

Drones use lightweight-client operation: they store block headers and verify inclusion proofs for transactions relevant to their assignments, reducing onboard storage/CPU demands.

3.6.2. Where matching is computed (off-chain)

To avoid expensive on-chain search and ranking, *matching is computed off-chain* at provider gateways using the latest signed **TxStatus** and **TxRequest** messages. The blockchain then records the resulting **TxAssign** and enforces confirmation/timeout/escrow logic. This preserves auditability while keeping on-chain workload bounded.

3.6.3. Interoperability with charging backends

Provider gateways can bridge session metadata to existing charging backends (e.g., OCPP-like operational workflows) while applying hardening guidance from charging protocol security studies [8]. Privacy-sensitive measurements can be protected using private metering extensions where appropriate [9].

This section specified a consortium-blockchain protocol for drone-assisted EV charging with (i) explicit roles and trust boundaries, (ii) a clear on-chain/off-chain data split with integrity anchoring, (iii) a message-level protocol and timing semantics, and (iv) implementable smart-contract workflows with escrow settlement and dispute handling. The next section analyzes security objectives, adversary capabilities, and protocol countermeasures in a structured threat model.

4. SECURITY FEATURES AND CHALLENGES

4.1. Threat Model and Security Objectives

Drone-assisted EV charging is a cyber-physical service spanning mobile endpoints (EVs and drones), provider gateways, and a consortium ledger. We consider the following adversary classes: (i) *external network attackers* attempting spoofing, replay, or denial-of-service; (ii) *malicious or compromised clients* (EV/drone) submitting false requests/status; (iii) *malicious insiders* (e.g., a provider operator) attempting billing manipulation; and (iv) *Byzantine validators* up to a bounded threshold under the consortium consensus assumptions. Security requirements for EV charging infrastructures and their backends include strong identity/authentication, metering/billing integrity, and resilience to network attacks [2, 8].

Accordingly, the protocol is designed to provide: **(1) Authentication and integrity** of all session-critical messages, **(2) non-repudiation and auditability** for assignment and settlement, **(3) availability under reasonable load and basic DoS attempts**, **(4) privacy-by-design** by minimizing on-chain sensitive data, and **(5) dispute tolerance** via escrow and verifiable completion evidence.

4.2. Data Integrity, Authentication, and Non-Repudiation

Digital signatures and replay protection: All session-critical transactions (request, assignment, accept/confirm, completion) are digitally signed by the originating party. Each signed payload includes a nonce and timestamp to mitigate replay and message reordering attacks. This addresses known

authorization and message-integrity weaknesses documented in EV charging ecosystems and protocol deployments [2, 8].

Tamper-evident ledger records: Confirmed on-chain events provide an immutable audit trail for who requested charging, which drone was assigned, whether both parties confirmed, and when escrow was settled. In a consortium/BFT setting, rewriting confirmed history requires collusion beyond the assumed Byzantine threshold (rather than a simple "51%" majority rule typical of PoW narratives).

Merkle inclusion proofs for lightweight clients: EVs and drones can operate as lightweight clients, verifying inclusion of their relevant transactions (e.g., assignment acceptance or settlement) using Merkle proofs without storing the full chain.

Integrity anchoring for off-chain telemetry: High-rate telemetry and detailed charging traces are stored off-chain (for privacy and scalability) while their integrity is anchored on-chain by committing cryptographic hashes and content pointers. This follows practical EV workflow designs that combine smart contracts with off-chain storage to reduce ledger bloat while preserving verifiability [12].

4.3. Authorization and Role-Based Access Control

Because the system spans multiple stakeholders (EV owners, drone operators, and providers), authorization must be explicit. The Registration/Authorization contract enforces role-based access control (RBAC) with least privilege: (i) EVs can submit charging requests and co-sign completion, (ii) drones can publish status and co-sign completion, (iii) providers/gateways can post assignment proposals and maintain off-chain repositories, and (iv) only consortium administrators (or a governance committee) can approve validator membership and parameter changes. This reduces the blast radius of compromised endpoints and aligns with approaches used in permissioned DLT EV assignment settings where compliance and misbehavior reduction are explicit goals [10].

To mitigate Sybil-style flooding, onboarding can require stakeholder approval and/or refundable deposits, and the Request contract can apply rate limits per identity.

4.4. Smart-Contract Security and Settlement Correctness

Escrow-based settlement: Payments are locked at acceptance time and released only after verifiable completion. Completion is evidenced by a dual-signature receipt (EV + drone) over session identifiers and delivered-energy summary, plus a hash pointer to the off-chain evidence bundle.

Timeouts and bounded liveness: Explicit timeouts (e.g., for assignment acceptance and for completion submission) prevent indefinite resource locking and reduce griefing vectors. Timeouts are critical in highly dynamic settings with intermittent connectivity (e.g., UAV-assisted charging contexts) [6].

Dispute handling: If the EV and drone completion receipts mismatch beyond a tolerance threshold, the contract transitions to a DISPUTE state. Resolution can be performed by a consortium-appointed arbitrator using the off-chain evidence bundle. Privacy-preserving reservation/payment designs emphasize that dispute handling must be part of the protocol rather than an afterthought [11].

Contract-level hardening: To reduce common smart-contract risks, contract code should avoid unbounded loops, apply checks-effects-interactions patterns for value transfers, use safe math, and emit events for auditable state transitions. Where feasible, static analysis and test harnesses should be used during development [12].

4.5. Drone-Specific Threats and Operational Countermeasures

Drone-assisted charging expands the attack surface beyond conventional station-based charging [2]. Key drone-specific risks include:

GPS spoofing / navigation manipulation: An attacker may attempt to misdirect drones or disrupt rendezvous. Operational mitigations include multi-sensor consistency checks (GPS + IMU/vision), geo-fencing, and accepting waypoints only when authenticated and policy-compliant. Because waypoint disclosure can leak sensitive locations, waypoint data should be encrypted in transit and minimized at rest.

Battery-drain and dispatch griefing: Adversaries may generate repeated requests or lure drones into costly routes. Mitigations include request deposits, reputation/rate limiting, bounded service radius constraints, and assignment acceptance windows that expire automatically.

False status updates: A compromised drone could advertise false capacity/location to win assignments. The protocol limits impact by requiring mutual confirmations and by anchoring key evidence (assignments, acceptances, completion receipts) on-chain; providers can additionally sanity-check drone reports against historical traces stored off-chain.

4.6. Availability, Latency, and DoS Resilience

Near real-time operation requires low-latency finality and resistance to overload. EV charging networks are known to face DoS and backend overload risks [2]. The protocol mitigates these risks via: (i) consortium low-latency consensus with fast finality, (ii) gateway batching of non-urgent updates, (iii) off-chain handling of high-rate telemetry, and (iv) request throttling and deposits to raise the cost of spam. Scalable on-/off-chain combinations are widely used in high-frequency transaction settings and motivate our partitioning design [14].

4.7. Privacy and Data Confidentiality

Charging traces and location histories can enable profiling. Privacy-preserving extensions for EV charging communications and billing are therefore essential [9]. Our protocol

TABLE 1: *Representative attack vectors, impacts, and primary mitigations in the proposed protocol.*

Attack Vector	Impact	Mitigation Strategy
Identity spoofing / request forgery	Unauthorized dispatch; billing fraud	Registration + RBAC; signed TxRequest/TxStatus; nonce/timestamp
Replay of confirmations	Incorrect acceptance/completion state	Nonce/timestamp; explicit state machine; timeouts
Sybil flooding	Queue overload; drone starvation	Onboarding controls; deposits; rate limiting; throttling
False status injection	Suboptimal or unsafe assignment	Mutual confirmations; provider sanity checks; auditable logs
Ledger inconsistency (Byzantine validators)	Double-spend / inconsistent settlement	BFT finality assumptions; consortium governance; validator monitoring
DoS on gateways or validators	Increased latency; dropped requests	Batching; backpressure; off-chain telemetry; throttling
GPS or navigation manipulation	Drone misrouting; session failure	Multi-sensor checks; geofencing; authenticated waypoint policy
Settlement disputes	Payment withheld or incorrect	Escrow; dual-signature completion; dispute state + off-chain evidence

applies: **Pseudonymous identities:** on-chain accounts are public keys rather than personal identifiers; **data minimization:** only coarse location and minimal session metadata appear on-chain; **encrypted off-chain records:** detailed traces are stored off-chain under access control, with on-chain hash anchors. Where stronger privacy is required, the protocol can incorporate privacy-preserving reservation/payment techniques (e.g., reducing linkability and enabling privacy-aware verification) [11, 13].

4.8. Scalability and Throughput Security Trade-offs

Scaling to many EVs/drones increases transaction volume and potential attack surface. Layer-2 channels and sharding can improve throughput, but introduce trade-offs (channel liquidity management, shard cross-communication, and more complex failure modes). Therefore, scalability choices must be evaluated together with adversary assumptions and operational constraints [14].

4.9. Balancing Security with Operational Efficiency

Security controls introduce overhead (signing, verification, consensus). The protocol reduces overhead by (i) keeping high-rate data off-chain with integrity anchors, (ii) using lightweight clients for drones/EVs, and (iii) limiting on-chain actions to session-critical events. This design aims to preserve auditability and resilience without imposing prohibitive latency for moderate-scale deployments [14].

This section defined a concrete threat model and detailed how cryptographic authentication, consortium finality, smart-contract escrow, and on-/off-chain partitioning jointly protect drone-assisted EV charging. The next section presents the coordination mechanism and shows how security constraints (timeouts, deposits, and confirmation workflows) integrate with assignment logic.

5. COORDINATION MECHANISM FOR EV CHARGING

5.1. Overview of Coordination Needs

Effective coordination in a drone-assisted charging network requires matching the right drone to the right EV at the right time while respecting mobility and resource limits. Key decision factors include:

- **Geographic proximity and travel time:** minimizing drone travel distance/time reduces service delay and operational energy.
- **Feasibility under energy/range constraints:** a drone must have sufficient remaining energy to reach the EV, deliver the requested charge (or minimum service), and maintain a safety reserve.
- **Service priority and fairness:** critical EVs (low state-of-charge, emergency fleet class, remote from fixed stations) should be served earlier, while mitigating gaming and request flooding.

These requirements align with prior work on decentralized EV assignment and cost-based allocation, which emphasizes feasibility constraints and fairness in multi-agent coordination [10].

5.2. Coordination Workflow and On-/Off-Chain Responsibilities

To keep on-chain workload bounded, **matching is computed off-chain** by provider-operated gateways (Coordinator Service), while the blockchain records assignments and enforces confirmation, timeouts, and settlement. The coordination proceeds in repeated rounds with a short batching window T_{batch} (typically 1–2 s under normal load; configurable under heavy load):

1. **Request intake:** EVs submit signed charging requests (TxRequest) with expiry and a maximum waiting time.
2. **State ingestion:** Drones periodically submit signed status updates (TxStatus) including coarse location and remaining deliverable energy.
3. **Off-chain matching:** the gateway computes priorities, filters infeasible drones, and selects assignments.
4. **On-chain commitment:** the gateway posts the selected assignments as TxAssign. The Assignment Contract verifies state constraints (request pending, drone available, not expired) and logs the assignment.

5. **Mutual confirmation:** EV and drone confirm within T_{confirm} (TxAcceptEV / TxAcceptDrone). Unconfirmed assignments expire automatically.

This division reduces smart-contract computation while preserving auditability and dispute-resilience through on-chain state transitions [14].

5.3. Priority Model for EV Requests

Each EV request i is assigned a normalized priority score $P_i \in [0,1]$. We use three normalized components:

- $\text{SoC}_i \in [0,1]$: EV battery state-of-charge (1 = full).
- $U_i \in [0,1]$: urgency level (e.g., derived from time-to-empty or service class).
- d_i^{st}: distance to nearest fixed charging station; normalized by a maximum service distance $d_{\max}$.

$$P_i = \alpha(1-\text{SoC}_i) + \beta U_i + \gamma \min\left(\frac{d_i^{\text{st}}}{d_{\max}}, 1\right) \tag{1}$$

where $\alpha, \beta, \gamma \geq 0$ and $\alpha+\beta+\gamma = 1$. This formulation is bounded, interpretable, and avoids unit inconsistencies.

5.4. Feasibility Filtering (Hard Constraints)

Before scoring, drones that cannot feasibly serve a request are filtered out. Let drone j have remaining deliverable energy E_j^{del} and current location x_j, and let the EV be at x_i. We define:

- $d_{ij} = \text{Distance}(x_i, x_j)$
- ETA_{ij}: estimated time of arrival based on cruise speed and optional congestion/no-fly constraints
- E_{ij}^{fly}: estimated flight energy to reach the EV (model-dependent; may be proportional to distance)

Drone j is feasible for request i only if:

$$\text{ETA}_{ij} \leq T_i^{\text{wait}} \tag{2}$$

$$E_j^{\text{del}} \geq E_i^{\text{req}} \cdot \eta_{\min} \tag{3}$$

$$E_j^{\text{bat}} \geq E_{ij}^{\text{fly}} + E^{\text{reserve}} \tag{4}$$

where T_i^{wait} is the EV maximum waiting time, E_i^{req} is requested energy (or minimum service energy), $\eta_{\min}$ is an efficiency/derating factor, and E^{reserve} is a safety reserve. These constraints prevent assignments that are physically implausible even if they look good under a purely proximity-based score.

5.5. Bounded Matching Score

For each feasible pair (i,j), we compute a bounded score $S_{ij} \in [0,1]$ that combines priority, proximity, and delivery capability:

$$S_{ij}^{\text{dist}} = 1 - \min\left(\frac{d_{ij}}{d_{\max}}, 1\right) \tag{5}$$

$$S_{ij}^{\text{eng}} = \min\left(\frac{E_j^{\text{del}}}{E_i^{\text{req}}}, 1\right) \tag{6}$$

$$S_{ij} = w_p P_i + w_d S_{ij}^{\text{dist}} + w_e S_{ij}^{\text{eng}} \tag{7}$$

where $w_p, w_d, w_e \geq 0$ and $w_p + w_d + w_e = 1$. This avoids the instability of $1/\text{Distance}$ and makes it easier to tune and reproduce.

5.6. Greedy Assignment Algorithm (Round-Based)

Let $\mathscr{Q}$ be the set of requests received during the last T_{batch} and $\mathscr{D}$ be the set of currently available drones. The gateway executes:

1. Compute P_i for all $i \in \mathscr{Q}$ using Eq. (1).
2. Sort requests by decreasing P_i (ties broken by earliest request timestamp).
3. For each request i in sorted order:
   (a) Build candidate set $\mathscr{D}_i \subseteq \mathscr{D}$ by feasibility checks Eq. (2).
   (b) If $\mathscr{D}_i$ is empty, keep i pending (or route to fixed infrastructure).
   (c) Otherwise choose $j^\star = \arg\max_{j \in \mathscr{D}_i} S_{ij}$.
   (d) Assign $j^\star$ to i, remove $j^\star$ from $\mathscr{D}$, and emit TxAssign on-chain.

This greedy-by-priority strategy is computationally lightweight and supports near real-time operation. More optimal global assignment (e.g., Hungarian/ILP) can improve total cost but may be too slow for large-scale high-frequency dispatch.

5.7. Smart-Contract Enforcement and Automation

Smart contracts automate and enforce only the **state transitions** needed for auditability and correct settlement:

- **Request lifecycle:** creation, expiration, cancellation, and closure (Request Contract).
- **Drone availability state:** available/en-route/charging and assignment locks to prevent double assignment (Assignment Contract).
- **Confirmation and timeouts:** EV and drone must confirm within T_{confirm}; otherwise the assignment expires automatically.
- **Settlement correctness:** payment is released only after TxComplete with **dual signatures** (EV+drone) and (optionally) a hash pointer to off-chain evidence (Payment/Escrow Contract).

5.8. Payment and Incentive Mechanisms

To align incentives with network objectives, the Payment Contract can incorporate: (i) a base fee proportional to delivered energy, (ii) a distance/time component (to reflect service cost), and (iii) a priority/remote-area bonus to ensure underserved regions remain serviceable. Auction-based and incentive-driven mechanisms for mobile charging have shown that incentive design materially affects allocation efficiency and provider participation [4, 5]. In addition, micropayment-oriented designs motivate automated settlement for repeated small-value charging events [15].

5.9. Coordination Flow Diagram

Fig. 1 illustrates the coordination workflow; matching and score computation occur at the gateway, while assignment, confirmation, and settlement are recorded and enforced on-chain.

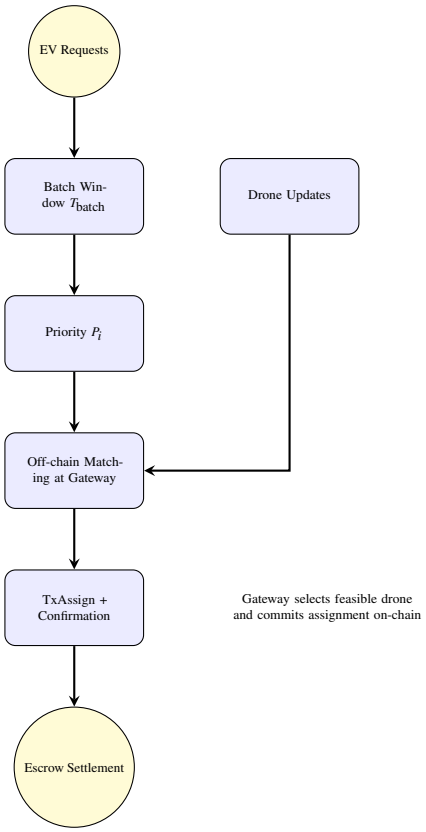

Fig. 1: *Coordination mechanism for matching EV charging requests with available drones (off-chain matching; on-chain enforcement).*

5.10. Resource Allocation Strategies

Table 2 summarizes common allocation strategies and trade-offs.

TABLE 2: *Comparison of resource allocation strategies for drone-assisted charging.*

Strategy	Key Features	Advantages & Disadvantages
Greedy Matching	Assigns a locally best drone per request	+ Simple, fast - May be suboptimal globally
Round-Robin	Cycles through drones uniformly	+ Equal usage - High travel cost
Priority-Based (ours)	Sort by P_i; choose best feasible drone by S_{ij}	+ Respects urgency and feasibility - Needs timely status updates
Cost-Minimization	Global optimization (LP/ILP) over costs	+ Near-optimal allocation - Higher compute cost

5.11. Algorithmic Complexity and Computational Feasibility

For each coordination round, greedy assignment runs in $O(|\mathscr{Q}| \cdot |\mathscr{D}|)$ time due to scoring feasible drone candidates per request, plus sorting $O(|\mathscr{Q}| \log |\mathscr{Q}|)$. This is suitable for real-time gateway execution. To further reduce runtime under scale, the search space can be reduced by geographic zoning, nearest-neighbor indexing, or hierarchical dispatch (local coordinators), consistent with scalable designs in large-scale wireless sharing economies [14].

5.12. Scalability Considerations

As the number of EVs/drones grows, coordination overhead can increase due to more frequent requests and updates. Two practical mechanisms are:

- **Partitioning the service area:** segment the region into zones with local drone pools to reduce candidate search space.
- **Hierarchical coordination:** local gateways perform immediate assignment; the consortium ledger records session-critical events to maintain global auditability.

This section specified a coordination mechanism consistent with a consortium-blockchain architecture: off-chain matching for low-latency feasibility-aware assignment, and on-chain enforcement for auditability, confirmations, and escrow settlement. The next section evaluates latency, throughput, and operational efficiency under representative loads.

6. PERFORMANCE EVALUATION

6.1. Methodology and Experimental Setup

We evaluate the proposed blockchain-based communication and coordination protocol using a combination of *discrete-event simulation* and *theoretical throughput/cost modeling*. The simulation captures EV and drone mobility, request arrivals, coordination and confirmation workflows, and network/consensus delays.

Simulation entities. EVs and drones are modeled as software agents with: (i) coarse location updates, (ii) battery/energy state variables, and (iii) protocol actions corresponding to TxRequest, TxStatus, TxAssign, confirmations, and completion/settlement events (Section 5 and Section III).

Workload model. Charging requests arrive according to a Poisson process with configurable intensity (normal vs. high load). To stress-test scalability, we also evaluate bursty traffic configurations that create large request backlogs (e.g., 200 requests entering the pending queue within a short interval), consistent with peak-hour demand spikes.

Network and consensus delay model. Communication delays are modeled as random latency on EV↔gateway and drone↔gateway links (configurable mean and variance), while blockchain confirmation is modeled using a fast-finality consortium setting with mean confirmation time of 5 s per block. Scalable designs in high-frequency transaction settings commonly adopt on-/off-chain partitioning and fast-finality configurations to reduce latency under load [13, 14].

On-chain vs. off-chain execution. To keep on-chain computation bounded, matching is executed off-chain at a provider gateway (Coordinator Service). The blockchain records session-critical events (assignment, confirmations, completion, settlement) and enforces state constraints and timeouts.

6.2. Reproducibility: Simulation Parameters

Table 3 summarizes the core parameters used in the evaluation. Unless otherwise stated, we report the mean over repeated simulation runs with different random seeds; stochastic variability can be reported as confidence intervals in an extended version.

6.3. Key Performance Metrics

We evaluate the system using the metrics in Table 4. To avoid ambiguity, we define **latency** as the elapsed time from the moment an EV request is received by the gateway to the moment the corresponding assignment is **confirmed** on-chain (including mutual acceptance where applicable). **Throughput** is measured as confirmed session-critical blockchain events per second (or completed charging sessions per second, depending on reporting granularity).

TABLE 3: *Core simulation and blockchain parameters used in performance evaluation.*

Parameter	Value / Description
Service area	50 km^2 region (uniform spatial distribution unless stated)
#EVs	100 (normal load), up to 200 (stress configurations)
#Drones	10
Drone flight range	25 km
Consensus confirmation time	5 s (consortium DPoS/BFT-style fast finality model)
Batching window	T_{batch} (short window; default 1–2 s in our settings)
Confirmation window	T_{confirm} (bounded acceptance window; configurable)
Request arrivals	Poisson process; optional burst injection for stress tests
Matching	Off-chain at gateway; on-chain enforcement via Assignment/Escrow contracts
Energy model	Drone energy use modeled proportional to travel distance (calibrated to Table 5)

TABLE 4: *Performance evaluation metrics for the proposed system.*

Metric	Description
Security Robustness	Attack success rate and service degradation under simulated adversarial actions
Latency	Mean time between request intake and assignment confirmation (seconds)
Throughput	Confirmed session events per second (or completed sessions per second)
Energy Efficiency	Drone energy consumed for dispatch/travel and protocol overhead
Resource Utilization	Fraction of drones actively assigned/serving over time
Cost Overhead	Estimated on-chain fees and compute overhead per completed session

6.4. Simulation Scenarios

We evaluate multiple configurations by varying area size, number of EVs, number of drones, and request intensity. A representative scenario includes:

- 100 EVs distributed across a 50 km^2 area,
- 10 drones with 25 km flight range,
- 5-second blockchain confirmation time in a delegated PoS consortium setting.

High-load scenarios are generated by increasing request intensity and/or injecting burst arrivals to create large pending queues (e.g., 200 requests waiting), allowing us to observe latency and stability under stress.

6.5. Comparative Baselines

We compare against:

1. **Centralized dispatch:** a single coordinator server performs matching and maintains an internal log. This baseline represents a high-performance but single-point-of-failure architecture.
2. **Non-blockchain decentralized:** endpoints use secure sockets (e.g., TLS) for communication, but coordination logs are not protected by an immutable ledger. This baseline reduces central dependency but lacks tamper-evident auditability.

To ensure a fair comparison, all approaches use the **same off-chain matching logic and mobility/workload models**; the key difference is whether assignment/confirmation/settlement events are recorded and enforced via consortium blockchain smart contracts.

6.6. Results and Analysis

6.6.1. Security Robustness

We evaluate security robustness by simulating adversarial behaviors commonly discussed for EV charging ecosystems (e.g., message manipulation, spoofing/replay attempts, and DoS-style overload) and by measuring: (i) attack success rate (whether an adversary can produce an accepted false assignment/settlement record), and (ii) service degradation (latency increase and failed sessions). In centralized and non-blockchain baselines, tampering with logs or manipulating messages can succeed if the coordinator/logging component is compromised. By contrast, in the proposed system, session-critical events are signed and committed to an auditable ledger; in our simulations we observed *zero successful ledger-tampering events* across thousands of sessions under the modeled attacker capabilities (i.e., attackers do not break cryptographic primitives, but can attempt spoofing, replay, and message alteration).

6.6.2. Latency and Throughput

As shown in Fig. 2, the proposed protocol maintains sub-10-second request-to-assignment confirmation latency under normal load, and remains competitive under high-load conditions. Two factors enable this behavior: (i) **off-chain matching** at gateways (reducing on-chain computation), and (ii)

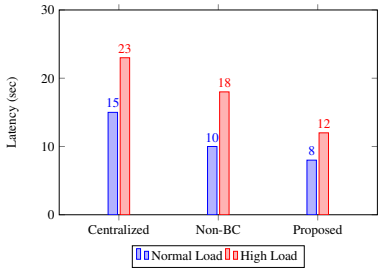

Fig. 2: *Latency comparison (average time for request-to-assignment confirmation) among three systems under normal and high load conditions.*

fast-finality consortium confirmation with short batching windows, which bounds end-to-end confirmation time.

Throughput (measurement and bound). We measure throughput as confirmed session events per second (and optionally as completed sessions per second). Additionally, given a mean confirmation time of 5 s, the *theoretical* maximum ledger throughput can be estimated from the configured block capacity as:

$$\mathrm{TPS}_{\max} \approx \frac{\text{tx per block}}{5}$$

In the tested scenarios, throughput demand remains below configured ledger capacity, and observed throughput is therefore dominated by service capacity (available drones) and workload intensity rather than by consensus saturation [13, 14].

6.6.3. Energy Efficiency

Drone energy consumption includes dispatch/travel and protocol overhead. Although periodic status updates introduce marginal communication overhead, improved assignment decisions reduce travel distance and idle time. Table 5 reports mean per-session values.

TABLE 5: *Energy metrics comparison (mean values per charging session).*

System	Energy Use (kWh)	Travel Distance (km)	Idle Time (minutes)
Centralized	4.2	12.5	18.0
Non-blockchain	4.1	12.1	16.5
Proposed (Blockchain)	4.0	11.8	15.0

The results show ≈5% reduction in per-session energy usage relative to the centralized baseline, primarily attributable to shorter travel distances and reduced idle time. In our simulator, travel energy is modeled as proportional to travel distance (consistent with the near-linear relationship between the first two columns in Table 5), which enables reproducible comparison across allocation strategies.

6.6.4. Cost Overhead

In a delegated proof-of-stake consortium setting, transaction fees are stable and comparatively low. In our evaluation, fees average approximately $0.01 per completed session. This per-session cost depends on the number of on-chain events recorded per session (assignment, confirmations, completion, settlement) and the configured fee schedule. Lightweight-client operation for drones reduces onboard storage and computation demands, and did not materially reduce effective flight duration in our simulation model.

6.7. Discussion and Limitations

Overall, the results indicate that consortium blockchain can improve auditability and resistance to tampering while remaining compatible with near real-time coordination when (i) matching is performed off-chain, (ii) the ledger records only session-critical events, and (iii) fast-finality consensus is used. However, limitations remain: performance depends on wireless connectivity and gateway availability, and large-scale deployments may require hierarchical dispatch, zoning, and additional off-chain scaling for very high request volumes [14].

7. CASE STUDY OR PRACTICAL APPLICATION

7.1. Urban Delivery Fleet Scenario

To illustrate the real-world viability of the proposed protocol, we consider a mid-sized urban delivery fleet. The fleet comprises 50 light commercial EV vans operating in a 30 km radius around a central warehouse. Range anxiety is a critical operational concern due to tight delivery schedules and varying travel routes. To mitigate this, the fleet operator deploys 8 drones capable of delivering up to 20 kWh each to stranded EVs or those running low on charge mid-route.

7.2. System Deployment

7.2.1. Setup and Onboarding

1. **Blockchain Consortium Formation:** The fleet operator collaborates with a local energy utility and a drone manufacturing firm. They form a consortium blockchain network, each entity running multiple validating nodes to ensure redundancy.
2. **Registration of Assets:** Each EV and drone registers its unique public key on the blockchain via the Registration Contract. Operational limits, such as maximum flight distance for drones, are also recorded.
3. **Mobile Application for Drivers:** The fleet operator integrates a smartphone application that interfaces with the blockchain, enabling drivers to request drone charging at the tap of a button.

7.2.2. Daily Operation

- **Route Monitoring:** A telematics system monitors each EV's battery level, distance traveled, and estimated time to depletion.
- **Charging Requests:** When an EV's battery level drops below a user-defined threshold (e.g., 25%), the application automatically sends a charging request to the blockchain.
- **Drone Dispatch:** The Assignment Contract evaluates available drones and dispatches the one with the highest matching score, considering proximity and drone battery reserves.
- **Encrypted Communication:** The drone and EV exchange encrypted GPS coordinates and authentication keys to ensure safe landing and hookup.

7.3. Performance Observations

- **Reduced Downtime:** EVs no longer have to detour to stationary charging stations. On average, each EV saved 15% in daily route time.
- **Enhanced Fleet Utilization:** Drones typically serviced 3–5 EVs per day. Over a week, the drones provided nearly 100 top-up charging sessions.
- **Secure Transactions:** Smart contracts automated the billing process, reducing administrative overhead. The blockchain ledger offered real-time visibility of transaction history among the three consortium partners.
- **Positive ROI:** After a six-month pilot, the operator reported that the cost savings from reduced EV downtime and streamlined logistics offset the initial capital expenditure on drones and blockchain infrastructure.

7.4. Challenges Encountered

1. **Regulatory Approval:** Local aviation authorities required detailed flight path logs and safety analyses for drone operations, extending the project timeline.
2. **Network Connectivity:** Certain urban areas with spotty 4G/5G coverage caused occasional delays in transaction confirmations. The consortium tackled this by installing edge servers in strategic locations.
3. **Battery Limitations:** Drones sometimes had insufficient capacity to fully recharge larger vans, necessitating multiple drones for a single session or partial top-ups.

7.5. Potential Extensions

- **Integration with Autonomous Driving:** Future expansions could enable fully autonomous vans to self-issue charging requests and even dispatch drones while on the move.
- **Smart Grid Interaction:** Collaborating with the energy utility to dynamically adjust drone charging schedules based on real-time grid loads, mitigating peak demand.

- **Predictive Analytics:** Leveraging AI-based forecasting models to predict charging demands, optimizing drone deployment routes ahead of time.

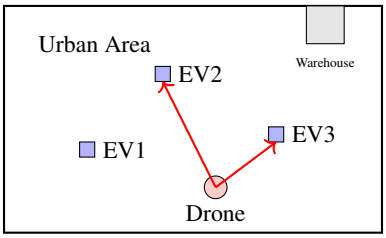

Fig. 3: *Schematic of an urban fleet scenario with drone-assisted EV charging.*

This case study demonstrates the practical implementation of a drone-assisted charging protocol powered by a consortium blockchain. The approach not only addresses range anxiety and reduces fleet downtime but also exemplifies how smart contracts and secure communication can streamline operations. Real-world challenges such as regulatory compliance and battery constraints highlight the need for ongoing refinement, yet the overall success underscores the potential for broader adoption in other commercial fleets and even private EV users.

8. CONCLUSION

The rapid evolution of electric vehicles and the parallel development of drone technologies have presented new opportunities to address the inherent challenges in conventional charging infrastructures. Fixed-location EV charging stations, despite their expanding presence, struggle to meet the demand for flexible, on-demand solutions—particularly in areas with sparse infrastructure or during peak usage times. This paper has introduced a blockchain-based secure communication and coordination protocol tailored for drone-assisted mobile EV charging, offering a scalable, transparent, and secure framework for the autonomous delivery of charging services.

By leveraging a decentralized ledger and smart contracts, the proposed system addresses critical security vulnerabilities such as data tampering, spoofing, and single points of failure. Our detailed review of security features demonstrates how cryptographic primitives and consensus mechanisms ensure data integrity and participant authentication throughout the charging process. The research further highlights a coordination algorithm that matches EV charging requests with suitable drones, optimizing for distance, battery constraints, and urgency. Simulated performance evaluations suggest that the system achieves a balance between security robustness, low latency, and high throughput. Compared to traditional centralized or non-blockchain solutions, the protocol exhibits superior resilience against adversarial actions and more efficient resource allocation.

A real-world case study of an urban delivery fleet underlines the pragmatic benefits of the system, showing reductions in both operational costs and downtime. While the case study reinforces the feasibility of drone-assisted charging, it also underscores challenges like regulatory compliance and drone battery limitations. The paper argues that ongoing technical improvements and regulatory harmonization are essential for mainstream adoption. Future research could delve deeper into AI-based optimization, multi-party integration with smart grids, and advanced cryptographic approaches such as zero-knowledge proofs and quantum resistance.

In conclusion, the synergy between drone mobility and blockchain's decentralized trust model holds substantial promise for advancing EV charging infrastructure to new levels of accessibility and reliability. By automating key tasks—ranging from charging session assignments to secure payments—the system reduces human errors, accelerates processes, and ensures transparent record-keeping for all stakeholders. As drone and EV technologies continue to mature, and as more robust regulatory frameworks take shape, blockchain-based solutions are poised to play a pivotal role in the transition to a more sustainable and flexible transportation ecosystem.

References

[1] A. R. Singh, S. Rangu, K. Reddy Madhavi, F. K. Alsaif, M. Bajaj, and I. Zaitsev, "Optimizing demand response and load balancing in smart EV charging networks using AI integrated blockchain framework," *Scientific Reports*, 2024.

[2] X. Hu, X. Jiang, J. Zhang, S. Wang, M. C. Zhou, B. Zhang, Z. Gan, and B. Yu, "Electric vehicle charging network security: A survey," *Journal of Systems Architecture*, 2025.

[3] A. Chowdhury, S. S. Shafin, S. Masum, J. Kamruzzaman, and S. Dong, "Secure electric vehicle charging infrastructure in smart cities: A blockchain-based smart contract approach," *Smart Cities*, 2025.

[4] Z. Husain, T. H. M. El-Fouly, S. Singh, R. Mizouni, H. Otrok, and E. F. El-Saadany, "Blockcharge: A blockchain-based auction framework for EV charging via mobile stations," *Applied Energy*, 2025.

[5] S. L. Hu, S.-H. Seo, K. Kang, and Q., "A blockchain-based electric vehicle charging cooperation model," *IEEE Transactions on Vehicular Technology*, vol. 74, no. 3, pp. 3941–3957, 2025.

[6] R. Xing, Z. Su, T. H. Luan, Q. Xu, Y. Wang, R. Li, and A. Benslimane, "UAVs assisted secure blockchain offline transactions for V2V charging among electric vehicles in disaster area," in *IEEE International Conference on Communications (ICC)*, 2022.

[7] A. Miglani, N. Kumar, A. Kishore, and N. Mohammad, "UAV-enabled edge computing and blockchain based secure charging station selection for energy trading in V2G

environment," in *Proceedings of the 5th International ACM MobiCom Workshop on Drone Assisted Wireless Communications for 5G and Beyond*, 2022, pp. 103–108.

[8] Z. Garofalaki, D. Kosmanos, S. Moschoyiannis, D. Kallergis, and C. Douligeris, "Electric vehicle charging: A survey on the security issues and challenges of the open charge point protocol (OCPP)," *IEEE Communications Surveys and Tutorials*, vol. 24, no. 3, pp. 1504–1533, 2022.

[9] N. F. Kamal, A. Sharida, S. Bayhan, H. Alnuweiri, and H. Abu-Rub, "Private metering in EV charging infrastructure: An OCPP extension," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 10, pp. 15 456–15 466, 2024.

[10] M. Moschella, P. Ferraro, E. Crisostomi, and R. N. Shorten, "Decentralized assignment of electric vehicles at charging stations based on personalized cost functions and distributed ledger technologies," *IEEE Internet of Things Journal*, 2021.

[11] S. M. Danish, M. M. Shabir, K. Zhang, H.-A. Jacobsen, and S. A. Hassan, "A blockchain-based privacy-preserving charging station reservation and payment scheme for electric vehicles," *Distrib. Ledger Technol.*, vol. 4, no. 3, 2025.

[12] S. Chaudhary, R. Gupta, R. Kakkar, S. Tanwar, Z. Polkowski, F. H. Alqahtani, and W. Said, "A smart contract and IPFS-based framework for secure electric vehicles synchronization at charging station," *Sustainable Energy, Grids and Networks*, 2024.

[13] D. Zhai, J. Liu, T. Zhang, J. Wang, H. Du, T. Liu, T. Wang, C. Zhang, J. Kang, and D. Niyato, "Epdb: An efficient and privacy-preserving electric charging scheme in internet of robotic things," *IEEE Internet of Things Journal*, 2024.

[14] T. C. Wan, W. Chen, K. E. Psannis, S. K. Goudos, Y. Yu, Z. Zheng, and S., "Scalable on-chain and off-chain blockchain for sharing economy in large-scale wireless networks," *IEEE Wireless Communications*, vol. 29, no. 3, pp. 32–38, 2022.

[15] D. Strugar, R. Hussain, M. Mazzara, V. Rivera, J. Y. Lee, and R. Mustafin, "On m2m micropayments: A case study of electric autonomous vehicles," in *2018 IEEE International Conference on Cybermatics*, 2018.

[16] K. Massmi, K. Hamouid, and K. Adi, "Secure electric vehicle dynamic charging based on smart contracts," in *International Symposium on Networks, Computers and Communications (ISNCC)*, 2023.

[17] N. S. Deshmukh, V. D. Khairnar, D. R. Vora, A. D. Jovanović, and F. Le Mouël, "Evmcsdlt: Electric vehicle mobile charging system using distributed ledger technology," *MethodsX*, 2025.