

# INNOVATIVE PROPOSAL FOR INTELLIGENT PARKING BASED ON BLOCKCHAIN

Raya Stoyanova<sup>1</sup>  
Filip Batalov

Received 04.05.2026.  
Revised 11.08.2026.  
Accepted 27.08.2026.

## Keywords:

Smart parking, Edge computing,  
Blockchain, Smart contract, Sensor  
fusion, Jetson Orin Nano

## ABSTRACT

*A conceptual multilayer Edge architecture for blockchain-based intelligent parking is proposed. Each parking unit combines a camera and supplementary sensors with an ESP32 or Arduino Mega controller. NVIDIA Jetson Orin Nano performs computer vision, sensor fusion, and decision-making locally. A permissioned blockchain network records reservations, state transitions, payments, and verifiable evidence through smart contracts. The time-critical actuation loop is separated from blockchain confirmation and remains local. The sensor vector, occupancy estimation, hysteresis, telemetry packets, authenticated commands, and communication latency are formalized. Video data, personal data, and detailed telemetry are stored off-chain. Only hashes and the minimum required business evidence are recorded in the distributed ledger. The proposed model provides a reproducible basis for prototyping an quantitative verification.*



© 2026 Published by Faculty of Engineering

## 1. INTRODUCTION

The growing number of vehicles and limited urban space make the search, reservation, and management of parking spaces a complex task. Centralized solutions usually integrate sensor data, user profiles, payments, and operator functions within a single infrastructure. This approach provides low latency. However, it also creates a single point of failure. It makes the independent auditing of reservations, tariffs, and parking-space state changes more difficult.

Blockchain technology enables the verifiable recording of business events and the execution of contractual rules through smart contracts. However, blockchain should not be included directly in the time-critical control loop for barriers, light indicators, and locking mechanisms. Network confirmation may be slow or temporarily

unavailable. Therefore, the present study combines local Edge processing with the asynchronous registration of verifiable events in a permissioned blockchain network.

The aim of the study is to develop a conceptual engineering model of an intelligent parking system. The model defines the hardware layers, signal exchange, functional allocation, parking-space states, and connection with the smart contract. The originality of the approach lies in its integrated architecture. It combines multimodal occupancy detection, local control, idempotent event exchange, minimized on-chain recording, and the formal separation of local response time from blockchain confirmation time.

The parking problem is part of the wider urban mobility system. The search for an available space increases the distance travelled. It creates additional conflict points and

<sup>1</sup> Corresponding author: Raya Stoyanova  
Email: [raikach@abv.bg](mailto:raikach@abv.bg)

places unnecessary demand on the road network. Economic models show that driver behaviour depends on location, price, expected parking duration, and the probability of finding an available space. Parking should therefore not be treated only as a local service. It influences traffic distribution, access to central urban areas, and the use of limited public space (Hagman, 2006; Shoup, 1997; Zakharenko, 2019).

When parking occupancy is high, individual searches develop into repeated circulation flows. These flows interact with transit traffic, public transport, freight deliveries, taxi services, and non-motorized road users. Peak-demand models and the macroscopic fundamental diagram show that local decisions concerning incoming traffic can affect the overall congestion level of the urban network (Aboudolas & Geroliminis, 2013; Arnott et al., 1993; Daganzo, 2007; Geroliminis, 2015; Haddad et al., 2013; Keyvan-Ekbatani et al., 2012; Vickrey, 1969). An intelligent parking system should therefore provide more than information about availability. It should also provide reliable mechanisms for reservation, pricing, access, and release of the parking space.

Current smart parking solutions use ultrasonic and magnetic sensors, cameras, RFID, mobile applications, cloud services, and forecasting algorithms. Their scope ranges from the detection of a single available space to integrated platforms serving multiple operators. The review by Fahim et al. (2021) shows that system effectiveness depends on detection accuracy, communication infrastructure, user interface, cost, and scalability. A centralized model is easy to manage. However, it places operational rules, payment records, and reservation histories under the control of a single administrative entity.

Blockchain has been investigated as a means of creating a shared and verifiable ledger among independent parking operators. Ahmed et al. (2019) proposed a multilayer architecture for integrating parking service providers. Their work emphasized trust and data integrity. Al Amiri et al. (2020) examined a consortium network and privacy protection during parking-space searches and reservations. Jennath et al. (2019) used smart contracts to manage relationships among landowners, operators, and users. Imbugwa et al. (2020) examined a mobile application based on Ethereum. These studies demonstrate the applicability of distributed ledgers. However, they use different assumptions concerning identity, payment mechanisms, and integration with physical sensors.

More recent studies have focused on combining blockchain, artificial intelligence, and digital twins. PARKTag integrates AI-based assessment and blockchain recording to support trusted and scalable management (Kalbhor et al., 2024). BSDP applies blockchain mechanisms and privacy protection within a digital-twin environment (Zhang et al., 2023). These

approaches extend system functionality. They also demonstrate the importance of an appropriate allocation of computational tasks. If every sensor signal becomes a blockchain transaction, system load and latency may become incompatible with the physical control of an entrance, barrier, or locking mechanism.

The unresolved engineering problem is how to combine three different temporal and trust domains. The first is the physical environment, in which the parking-space state must be identified quickly and reliably. The second is the operational information system, which processes reservations, access rights, and tariffs. The third is the blockchain ledger, which provides verifiability among participants with different interests. The absence of a clear distinction among these domains leads either to excessive centralization or to the unjustified transfer of time-critical operations to the distributed ledger.

The present study addresses this gap through an Edge–blockchain framework. Physical decisions are made locally, while verifiable business events are recorded asynchronously. The scientific contribution includes the functional separation of the local and contractual control loops. It also includes a formal model for sensor fusion, hysteresis-based state determination, idempotent event exchange, a limited on-chain dataset, and measurable indicators for future verification. The proposal therefore extends beyond a general description of blockchain benefits. It provides a reproducible engineering specification.

## **2. METHODOLOGY**

The study applies conceptual engineering design, functional decomposition, and formal modelling. The components are allocated according to their computational resources, required response time, auditability requirements, and the sensitivity of the processed data. The applicability of the architecture is assessed by tracing the main operational scenarios. These include monitoring, reservation, entry, parking, payment, exit, and recovery after a communication failure. Permitted state transitions and potential failure modes are also analysed.

The proposed development is conceptual and does not currently include a physical prototype. Therefore, the quantitative characteristics are defined as measurable design indicators and criteria for future experimental verification. The formal model specifies the variables and relationships that should be measured during the development of a test bench.

The methodological process consists of five consecutive stages. First, the system participants and functional scenarios are defined. The participants include the driver, parking operator, municipal authority, payment service provider, and technical administrator. Second, the physical and information events are identified. Third, the

functions are allocated among the parking-space controller, Edge node, off-chain database, and blockchain network. Fourth, a state machine and a set of smart contract functions are developed. Fifth, indicators for accuracy, latency, reliability, throughput, and resource utilization are defined.

The main scenarios are represented as traceable chains of cause, decision, and evidence. During a reservation, the system verifies the contractual availability of the parking space and the permitted time window. At the entrance, the local node validates the authorization and the physical presence of the vehicle. During parking, the sensors confirm occupancy. At the exit, the fee is calculated and the payment is validated. The parking space is returned to the FREE state only after its physical release has been confirmed. Each scenario considers normal operation, communication failure, duplicate messages, sensor conflicts, and discrepancies between the physical and contractual states.

The architectural design is evaluated according to the principles of least privilege, separation of responsibilities, secure-by-default operation, and recoverability. The parking-space controller can execute only a limited set of commands. The Edge node does not store unencrypted payment credentials. The smart contract does not directly control electromechanical devices. The blockchain nodes are maintained by previously identified organizations. This model is more appropriate for municipal or multi-operator infrastructure than a public blockchain network with anonymous validators.

MQTT over TLS is selected as the main communication protocol. Its publish-subscribe model separates telemetry producers from telemetry consumers (OASIS, 2019). Quality of Service level 1 provides at-least-once delivery, but duplicate delivery remains possible. Therefore, reliability does not depend only on the communication protocol. It is supported by an eventId, a sequenceNo, a timestamp, and verification of the application state. RS-485/Modbus RTU is provided as an alternative for wired industrial implementations. HTTPS REST or gRPC may be used between the Edge gateway, application server, and protected off-chain services.

The permissioned blockchain network is conceptually compatible with Hyperledger Fabric (Hyperledger Foundation, n.d.). This choice is based on its managed identities, endorsement policies, and separation of transaction execution, ordering, and validation. The specific blockchain platform is not a mandatory element of the scientific model. The same functions can be implemented in another permissioned environment. Such an environment must provide organizational authentication, deterministic execution, access control, and transaction finality.

The evaluation criteria are defined in advance. This prevents design assumptions from being presented as measured results. Detection performance is evaluated through accuracy, precision, recall, and the F1-score. Communication latency is evaluated using its median, 95th percentile, and 99th percentile. The blockchain layer is assessed through time to finality, transactions per second, failure rate, and ledger size. System resilience is evaluated using recovery time, lost events, duplicate events, and state inconsistencies.

**Declaration on the use of generative artificial intelligence:** Generative AI tools were used for language editing, translation support, and improvement of textual clarity. In addition, Figure 1 was digitally rendered with the assistance of a generative AI tool, based on an original sketch and technical specifications prepared by the authors. The authors determined the scientific concept, system architecture, relationships between the components, labels, and final composition of the figure and verified its technical accuracy. The authors developed and take full responsibility for the scientific content, mathematical model, analysis, results, and conclusions of the manuscript. No AI tool was used to generate research data or scientific results.

### 3. RESULTS OF THE CONCEPTUAL DESIGN

The proposed architecture has a hierarchical structure. It includes a sensor layer, parking-space controller, Edge node, communication layer, permissioned blockchain network, application layer, and local actuation loop. The information flow moves from physical observation to the local decision and the blockchain ledger. The control flow returns to the parking-space controller and the actuating device. The overall configuration of the system layers is presented in Figure 1.

#### 3.1 System Architecture Sensor Node and Parking Space Controller

Each parking unit includes an IP camera and one or more supplementary sensors. These may include an ultrasonic sensor, a magnetic sensor, or a barrier-status sensor. ESP32 is suitable for a wireless prototype implementation. Arduino Mega may be used when more inputs and outputs or an external communication module are required. The controller samples the signals and verifies their permissible ranges. It also synchronizes time, generates a telemetry packet, and maintains a local buffer during communication failures.

The parking unit rejects clearly invalid measurements and marks diagnostic events. It accepts only authenticated commands with a valid expiration time and a new nonce. If communication is lost, a predefined safe mode is activated. Feedback from the barrier, locking mechanism, or light indicator confirms the physical execution of the command.

### 3.2. NVIDIA Jetson Orin Nano Edge Coordinator

NVIDIA Jetson Orin Nano (NVIDIA Corporation, n.d.) acts as the intermediate intelligent node for a parking sector or the entire parking facility. It receives data packets from the individual parking units and processes the video stream locally. It performs vehicle or licence-plate recognition when a valid legal basis exists. The device fuses the results from the camera and supplementary sensors. It also evaluates the confidence of the detected state. The local database stores the current operational state, safety rules, permitted actions, and pending events.

The Edge node generates a transaction only after a local decision has been made. An outbox queue stores unconfirmed events. It retransmits them when the connection is restored. A unique event Id and a monotonically increasing sequence No provide idempotency. They prevent the same event from being applied more than once.

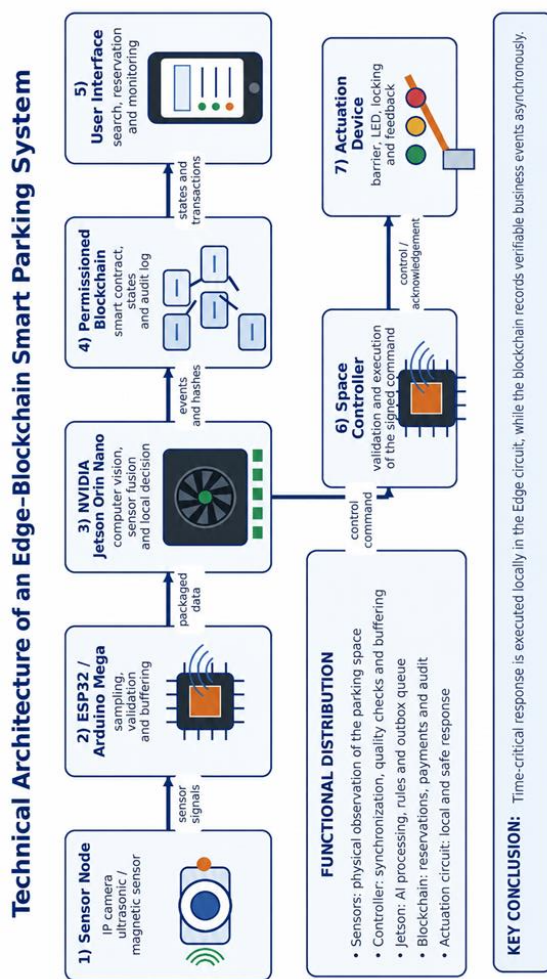


**Figure 1.** Technical architecture of the proposed system

Note: Digitally rendered with the assistance of generative AI, based on an original sketch and technical specifications prepared by the authors

### 3.3. Communication and Blockchain Layer

Telemetry data and commands can be exchanged through MQTT over TLS. Each parking unit publishes data to its own topic. Jetson is authenticated as a gateway and publishes digitally signed commands. QoS 1 provides at-least-once delivery. The application therefore verifies the event Id and sequence No and rejects duplicate packets. RS-485/Modbus RTU may be used between the parking units and the Edge node where long cable runs or electromagnetic interference are present.

The permissioned blockchain network manages participant identities, smart contracts, reservations, tariff versions, payment references, and an audit log. Video frames, licence-plate numbers, complete payment data, and detailed telemetry remain off-chain. The ledger records the parking-space identifier, pseudonymized identifiers, timestamps, event hash, previous state, new state, and transaction identifier.

Functional decomposition produces a seven-layer architecture. The sensor layer monitors the physical environment. The parking-space controller performs signal sampling, initial validation, and management of local inputs and outputs. The Edge layer fuses the data and makes the operational decision. The communication layer provides secure data transmission. The permissioned blockchain stores verifiable business events. The application layer provides interfaces for drivers and operators. The actuation loop controls the barrier, indicator, or locking device. This allocation limits the propagation of failures between the system layers.

Two related but distinct models are maintained for each parking space. The physical model is based on observation and may have the states

FREE, OCCUPIED, UNCERTAIN, and  
SENSOR\_FAULT.

The contractual model represents rights and obligations. It uses the states

FREE, RESERVED, OCCUPIED,  
PAYMENT\_PENDING, and  
OUT\_OF\_SERVICE.

This distinction is important. A sensor may detect a vehicle without a valid reservation. Conversely, the smart contract may temporarily maintain a reservation while the space remains physically unoccupied. The system does not conceal this discrepancy. It records it as a diagnostic case.

Sensor fusion reduces dependence on a single measurement principle. The camera provides detailed spatial information. However, it is sensitive to lighting, contamination, and obstruction. The ultrasonic sensor measures distance. Its readings may be affected by geometry and atmospheric conditions. The magnetic

sensor responds to the vehicle's metal mass. Its output depends on installation conditions and vehicle type. Combining these sources enables the system to evaluate both the state and the confidence of the decision.

Edge processing reduces the need for continuous video transmission to a central cloud. The camera stream is analyzed within a protected local environment. Only events and a minimum set of features are transmitted to the other components. A video frame may be retained only for a predefined purpose, such as a disputed event or diagnostic analysis. It must also be retained for a limited period. This approach reduces network traffic and the exposure of personal data. However, it requires management of the recognition models, software versions, and local detection quality.

The outbox mechanism is essential for a reliable connection between the local database and the blockchain network. After a validated physical event, the Edge node stores the business event and its status in a local transaction. A separate process forwards the event to the blockchain gateway. A confirmed event is marked as completed. An unconfirmed event remains in the queue. During retransmission, the smart contract verifies the unique identifier. It does not apply the same change twice. A temporary communication failure therefore neither blocks the parking facility nor produces duplicate charges.

The command channel uses a different model from the telemetry channel. Each command contains a device identifier, permitted action, time window, nonce, policy version, and digital signature. The controller verifies the signature, validity period, sequence, and admissibility of the command against the current local state. An OPEN command is not executed if a limit switch indicates a mechanical fault or if safe mode is active. After execution, a separate acknowledgement confirms the physical result.

Identity management covers individuals, organizations, applications, and devices. A user is identified at the application layer. A pseudonymized identifier is used in the blockchain. The parking operator may publish availability data and tariff versions for its own facilities. A municipal authority may have auditing rights without permission to alter commercial records. The payment service provider confirms the payment reference without disclosing complete card details. Devices receive time-limited certificates that can be revoked.

The tariff is represented as a versioned set of rules. It contains a base price, time unit, permitted grace period, maximum daily charge, user categories, and validity period. Dynamic changes are allowed only for future parking sessions or under a previously published rule. The hash of the tariff version is recorded when the parking session begins. The subsequent fee calculation

can therefore be verified. The operator cannot replace the applied tariff without leaving evidence of the change.

Separate interfaces are provided for the driver, operator, and auditor. The mobile application displays available spaces, prices, conditions, and the reservation period. The operator interface displays system states, alarms, pending events, and device health. The audit interface enables a transaction chain to be traced through pseudonymized identifiers and hashes. It does not provide unrestricted access to video recordings or payment data. All administrative actions are recorded separately.

System security is not limited to the immutability of the ledger. A compromised sensor may generate a false event that the blockchain will correctly record as received. The system therefore uses mutual authentication, secure boot, signed software updates, network segmentation, configuration-change logging, and anomaly monitoring. If a significant discrepancy occurs between the camera and another sensor, the parking space enters the UNCERTAIN state. It is not automatically charged or released.

### **3.4. State Model and Smart Contract Logic**

The physical state of a parking space is determined by the sensors. Its contractual state is managed by the smart contract. The main contractual states are

FREE, RESERVED, OCCUPIED,  
PAYMENT\_PENDING, and  
OUT\_OF\_SERVICE.

The normal sequence is

FREE → RESERVED → OCCUPIED →  
PAYMENT\_PENDING → FREE.

An expired or cancelled reservation returns the state from RESERVED to FREE. An administrative or diagnostic event may move any permitted state to OUT\_OF\_SERVICE. The smart contract implements the functions register User, register Vehicle, query Available Spaces, reserve Space, confirm Entry, start Session, calculate Fee, confirm Payment, confirm Exit, cancel Reservation, and set Out Of Service. Each function has preconditions, an authorized caller role, time constraints, and a defined result. For example, reserve Space is executed only if the parking space is FREE and the user has been validated. Confirm Entry requires an active reservation and correspondence between the vehicle identifier and the authorization. Confirm Exit releases the parking space after the session has ended and the payment or another permitted arrangement has been validated.

A discrepancy between the physical and contractual states is not corrected automatically through a potentially unsafe action. Instead, it generates a diagnostic event. The event initiates another verification, a safe local

response, and operator intervention. The smart contract therefore manages the business rules but does not replace the local safety logic.

The smart contract logic is determined by the caller's role, current state, time, and event uniqueness. The reserveSpace function verifies that the space is FREE, the tariff version is active, and the user has no conflicting reservation. If these conditions are met, it creates a reservation Id, a start time, an expiration time, and a Reservation Created event. A repeated call with the same request Id returns the previously created result. A concurrent request for the same parking space is rejected after the first valid change to the state version.

The confirm Entry function is called by an authenticated Edge gateway after local verification. It links the reservation Id, pseudonym zed vehicle identifier, and hash of the physical evidence. The start Session function records the starting time and tariff version. The calculate Fee function is a pure deterministic function based on the parking duration and applicable rules. The confirm Payment function accepts only a payment reference and status from a trusted payment adapter. The confirm Exit function ends the session. However, the release of the parking space is recorded as final only after confirmation from the physical layer.

Exceptions are an integral part of the smart contract. The cancel Reservation function applies different rules before and after the reservation start time. The expire Reservation function is activated by an authorized scheduler and releases an unused reservation. The set Out Of Service function requires an administrative role and a stated reason. The reconcile State function does not arbitrarily alter the existing history. It creates a new corrective event linked to the original record. Disputes may be examined through the evidential sequence. The legal decision remains outside the automated contract.

Input validation, prevention of unauthorized repeated calls, role control, contract versioning, and an emergency pause are provided to limit vulnerabilities. An upgrade must not erase the previous rules. Each version has its own identifier and period of validity. Before deployment, the functions should undergo unit testing, property-based testing, and concurrent transaction scenarios. Critical rules should also be subjected to an independent code review (Khan et al., 2021).

### 3.5. Mathematical Model of Signal Exchange

Let  $i = 1, 2, \dots, N$  denote the parking-space identifier and  $k$  the discrete observation time. After preprocessing, the controller generates a feature vector. It combines the camera-based occupancy probability, distance measured by the ultrasonic sensor, magnetic response, and feedback from the actuating device:

$$x_i[k] = \quad (1)$$

$$= [p_i^{cam}[k], d_i^{us}[k], m_i^{mag}[k], b_i^{bar}[k]]^T$$

The components have different physical dimensions and ranges. To make them comparable, each feature is standardized using the mean value and standard deviation determined during calibration:

$$\tilde{x}_{i,j}[k] = \frac{x_{i,j}[k] - \mu_{i,j}}{\sigma_{i,j}}. \quad (2)$$

The occupancy probability is obtained through the weighted fusion of the normalized features. The weights may be determined through training or expert calibration:

$$p_i^{occ}[k] = \sigma_i(w^T \tilde{x}_i[k] + \beta), \quad \sigma(z) = \frac{1}{1 + e^{-z}}. \quad (3)$$

Two thresholds,  $(\theta_{off} < \theta_{on})$ , are introduced to prevent rapid switching near the boundary values. A hysteresis law is used to determine the physical state:

$$S_i^{phys}[k] = \begin{cases} OCCUPIED, & p_i^{occ}[k] \geq \theta_{on}, \\ FREE, & p_i^{occ}[k] \leq \theta_{off}, \\ S_i^{phys}[k-1], & \theta_{off} < p_i^{occ}[k] < \theta_{on}. \end{cases} \quad (4)$$

The controller transmits a structured packet instead of a raw video stream. A separate secure channel may be used when video is required for local analysis. The general telemetry packet contains the schema version, identifiers, timestamp, sequence number, features, state, quality indicators, and hash:

$$\mathcal{P}_i[k] = \{v, deviceID_i, slotID_i, t_k, seq_i, x_i[k], S_i^{phys}[k], q_i[k], h_i[k]\}. \quad (5)$$

Cryptographic evidence of the packet content is generated by concatenating the critical fields and applying SHA-256:

$$h_i[k] = \text{SHA} - 256(v || deviceID_i || \times slotID_i || t_k || seq_i || S_i^{phys}[k] || q_i[k]). \quad (6)$$

The uplink latency extends from the physical measurement to the validated Edge state. The downlink latency extends from the Edge decision to physical execution:

$$\tau_i^\uparrow = \tau_{sens} + \tau_{ctrl} + \tau_{net}^\uparrow + \tau_{edge}, \quad \tau_i^\downarrow = \tau_{dec} + \tau_{net}^\downarrow + \tau_{ctrl} + \tau_{act}. \quad (7)$$

Jetson generates a signed command containing the permitted action, validity period, nonce, and signature. The controller returns an acknowledgement. This acknowledgement links the command to its actual physical execution

$$\begin{aligned} \mathcal{C}_i[k] &= \{\text{cmdID}, \text{slotID}_i, u_i[k], t_{\text{exp}}, \text{nonce}, \text{sig}\} \\ \mathcal{A}_i[k] &= \{\text{cmdID}, \text{status}, t_{\text{exec}}, h_i[k]\}. \end{aligned} \quad (8)$$

The time required for final confirmation in the permissioned blockchain network includes propagation, endorsement, ordering, validation, and commitment

$$\tau_{\text{BC}} = \tau_{\text{prop}} + \tau_{\text{endorse}} + \tau_{\text{order}} + \tau_{\text{validate}} + \tau_{\text{commit}}. \quad (9)$$

The time-critical loop remains independent of blockchain latency. With asynchronous recording, the local response time is determined only by the uplink and downlink Edge channels:

$$\begin{aligned} \tau_{\text{response}} &= \tau_{\text{local}} = \tau_i^{\uparrow} + \tau_i^{\downarrow} \leq \tau_{\text{max}}, \\ \frac{\partial \tau_{\text{response}}}{\partial \tau_{\text{BC}}} &= 0 \end{aligned} \quad (10)$$

The process diagrams from the original model complement the system architecture. Figure 2 presents preliminary registration and reservation. Figure 3 shows the entry sequence. Figure 4 describes the exit procedure and smart contract finalization. These diagrams remain unchanged as a functional specification. The technical extension clarifies which operations are performed locally, which are executed by the application layer, and which generate blockchain transactions.

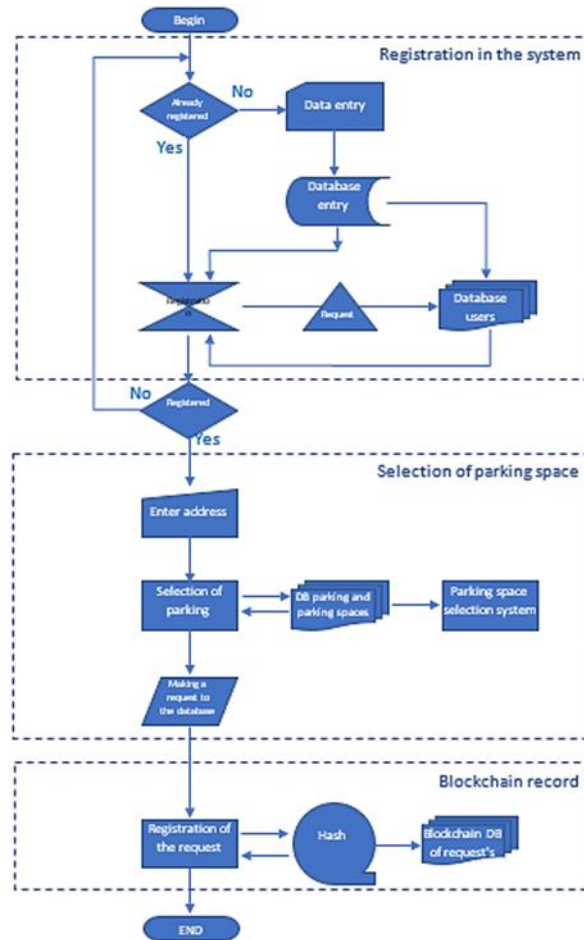


Figure 2. Algorithm for preliminary registration reservation and smart contract creation

### 3.6. Design Indicators and Verification Plan

Experimental verification requires a test bench containing representative sensors, an ESP32 or Arduino Mega controller, NVIDIA Jetson Orin Nano, an actuating device, and a permissioned blockchain network. The testing plan includes the following measurable indicators:

- accuracy, precision, recall, and F1-score when detecting available and occupied parking spaces under different lighting conditions, partial obstruction, and sensor failures;
- local elapsed time from a sensor event to command generation and confirmed physical execution;
- elapsed time from transaction generation to final blockchain confirmation;

- throughput during concurrent reservations and the proportion of failed transactions;
- correct operation when messages are duplicated, delayed, or received out of sequence;

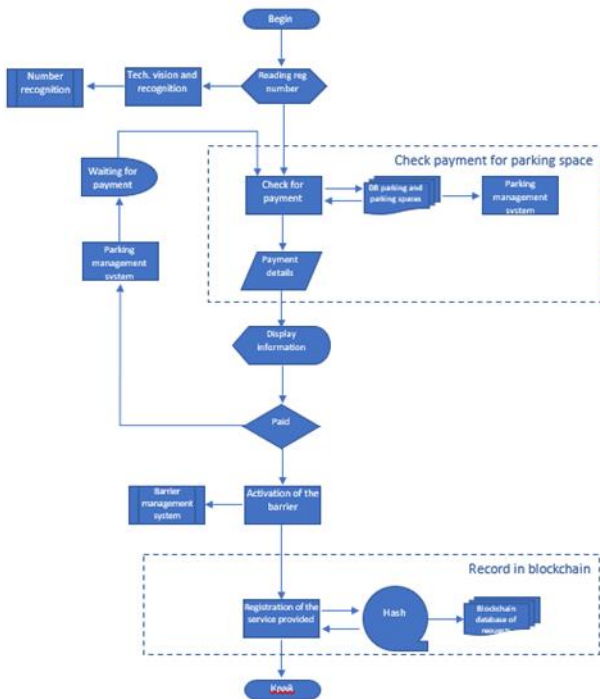


**Figure 3.** Algorithm for automated entry into the parking facility

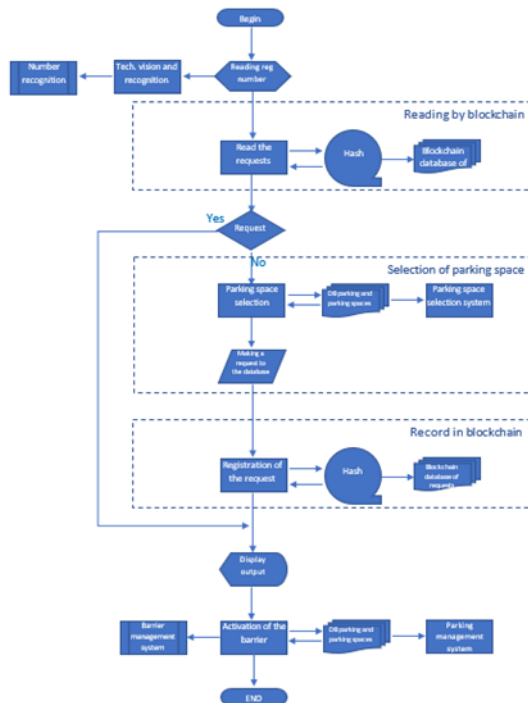


**Figure 4.** Algorithm for exit and smart contract finalization

- recovery after failure of a controller, Edge node, or communication link;
- CPU, GPU, memory, network, and energy utilization of the Edge node;
- volume and growth rate of the on-chain and off-chain data.

The workload, number of parking spaces, event frequency, network conditions, software version, and number of repeated trials are specified for each test. Separate measurement of local latency and blockchain confirmation time enables the condition in Equation (10) to be verified. It also makes it possible to determine whether the safety-related action remains independent of the distributed ledger state.

The test plan includes at least three groups of operating conditions. Normal conditions include daytime and nighttime illumination, different vehicle categories, and standard network loads. Boundary conditions include partial obstruction, a contaminated camera, measurements close to the decision threshold, and vehicles entering in rapid succession. Failure conditions include interruption of the MQTT connection, restarting the Edge node, an unavailable blockchain gateway, duplicate packets, and failure of the actuating device. Each scenario is repeated several times using a fixed system configuration.

The number of virtual parking spaces and the event frequency are increased gradually to evaluate scalability. The outbox queue length, processor and memory utilization, MQTT broker load, endorsement and commitment times, and blockchain-ledger size are measured. Peak intervals are simulated separately. During these intervals, multiple drivers search for and reserve a limited number of parking spaces. An acceptable configuration must maintain the local response within the specified threshold. It must also process contractual events without data loss.

The cost assessment should distinguish between the initial investment and operating costs. The initial investment includes sensors, controllers, cameras, Edge nodes, communication equipment, and installation. Operating costs include electricity, connectivity, maintenance, certificates, software updates, data storage, and operation of the blockchain nodes. The permissioned model does not involve unpredictable public-network gas fees. However, it generates organizational costs associated with participant and infrastructure management.

The acceptance criteria must be defined according to the specific application. The required local response time for a barrier in an enclosed parking facility may be stricter than the update time of the user interface. The loss of a payment event is unacceptable. In contrast, a short delay in an analytical report may be tolerated. The proposed design therefore does not impose a single universal

numerical threshold. It provides a measurement method and classifies the indicators according to safety, operational service, auditing, and analytics.

#### **4. DISCUSSION**

The proposed architecture combines fast local response with verifiable recording of business events. Edge processing limits the transmission of raw video data. It reduces dependence on external connectivity and enables sensor fusion close to the data source. The permissioned blockchain network reduces the risk of unauthorized changes to reservations, tariffs, and payment references. It also provides a shared audit log for the participating organizations.

These advantages introduce additional complexity. The system requires management of digital identities and cryptographic keys. It also requires access policies, synchronization between local and contractual states, monitoring of the outbox queue, and recovery procedures. Blockchain does not guarantee the validity of primary sensor data. It provides evidence of what was recorded and by whom. Measurement quality, calibration, diagnostics, and protection of Edge devices therefore remain critical.

Minimizing on-chain data is necessary for both scalability and personal data protection. Video recordings, licence-plate numbers, and payment data should be processed only when a specific legal basis exists. They should have a limited retention period and remain within a controlled off-chain environment. The blockchain ledger should contain only the evidence required to verify the system state and transaction.

##### **4.1. Comparison with Existing Solutions**

Compared with a conventional centralized system, the proposed framework provides shared verifiability among independent operators. This is useful when a municipality, private parking operators, and payment service providers must follow common rules. No single participant can unilaterally alter the recorded history. The corresponding disadvantage is more complex management of identities, policies, and software versions. If the system has only one operator and does not require independent auditing, a conventional secure database may be a more economical solution.

The architecture proposed by Ahmed et al. (2019) integrates multiple service providers through blockchain. The present model provides more detail about the connection with physical control. It also separates the time-critical Edge loop from the blockchain layer. Parkchain applies smart contracts to the use of unoccupied land and the distribution of revenue (Jennath et al., 2019). In the present study, the main object is the lifecycle of a specific parking space. The model establishes a verifiable relationship among the sensor

event, reservation, tariff, and payment. PARKTag combines artificial intelligence and blockchain. The present contribution additionally formalizes idempotency and distinguishes between local and blockchain latency (Kalbhor et al., 2024).

Privacy-oriented solutions demonstrate that an immutable record alone is insufficient. Al Amiri et al. (2020) applied private information retrieval and anonymous reservation mechanisms. BSDP also emphasizes the protection of user requests and dynamic data (Zhang et al., 2023). The proposed framework follows a more pragmatic data-minimization principle. Sensitive data remain off-chain. The ledger contains pseudonyms, hashes, and business events. This approach does not provide complete anonymity. However, it reduces data exposure and facilitates compliance with organizational policies.

##### **4.2. Effects on Traffic Flows and Tariff Management**

Reliable information about parking availability and the possibility of advance reservation may reduce unnecessary searching near the destination. However, the effect depends on data accuracy and driver behaviour. Incorrect information about an available space may increase rather than reduce the distance travelled. The transport effect should therefore not be inferred solely from the implementation of a mobile application. It should be evaluated through the average search time, additional distance travelled, entrance traffic volumes, and changes in circulation flows.

Tariff management can support a more balanced use of the available parking spaces. Publishing tariff versions and automatically applying previously disclosed rules can reduce disputes. However, dynamic pricing must be transparent and predictable. The algorithm must not change the price after the parking session has begun. It must not use protected personal characteristics. Appropriate factors include the zone, time period, observed occupancy, and officially defined vehicle category.

Integration with urban systems may provide aggregated data for planning. However, the blockchain ledger should not become a repository for all transport observations. Traffic management requires aggregated and time-sensitive indicators. Auditing requires a limited set of evidential records. Separating these purposes reduces the number of blockchain transactions. It also allows analytical models to develop independently of the smart contract logic.

##### **4.3. Technical Feasibility Scalability and Costs**

Practical implementation may begin with a pilot sector containing a limited number of parking spaces. One Edge node can serve a group of parking-space controllers and

publish data to a secure broker. The blockchain network may initially include the parking operator and a municipal organization. Other participants may be added later. This staged approach allows detection accuracy and operational reliability to be verified before expansion. It also limits the risk of investing in expensive infrastructure without demonstrated benefits.

System throughput is managed through aggregation and event selection. Periodic diagnostic measurements remain off-chain. The blockchain records the reservation, confirmed entry, start and end of the session, tariff version, payment reference, significant administrative actions, and hashes of disputed evidence. Some events may be grouped when this does not compromise traceability. The transaction rate is therefore determined by the business processes rather than the sensor sampling frequency.

Transaction processing time is not the same as the time required to open the barrier. The local decision uses previously synchronized permissions and rules. Blockchain confirmation follows asynchronously. If the network is unavailable, the system operates under a restricted offline policy and stores the events in the outbox. Prolonged unavailability may restrict new reservations. However, it must not prevent users from leaving the parking facility safely.

Economic feasibility depends on the scale of the system and the number of participating organizations. Additional blockchain infrastructure is justified when the value of verifiability, automated revenue allocation, and a shared ledger exceeds the cost of nodes, management, and maintenance. A centralized system may achieve the same practical result for a small private parking facility. The framework therefore includes a negative selection criterion. Blockchain should not be used when there is no multi-party trust problem.

#### **4.4. Limitations and Future Work**

The main limitation is the absence of a physical prototype and measured experimental indicators. The equations provide a design basis. They do not report achieved performance. The actual values will depend on the camera, sensors, recognition model, message broker, network topology, and blockchain platform configuration. This limitation is significant. At this stage, it is not possible to claim quantitative superiority over other systems.

A second group of limitations concerns the reliability of data from the physical environment. Blockchain prevents an undetected alteration of a recorded event. However, it cannot prove that the sensor performed the measurement correctly. Calibration, diagnostics, periodic inspections, and secure device management are therefore required. A third limitation concerns the legal and organizational management of identities, video data, and payment

references. These requirements must be specified for the relevant jurisdiction and parking operator.

Future work includes the development of a test bench and the training and evaluation of a detection model. It also includes load testing of MQTT and the permissioned blockchain, failure analysis, and evaluation of the total cost of ownership. A pilot implementation in an operational parking facility should follow the laboratory tests. It should compare a baseline centralized configuration with the proposed Edge-blockchain configuration. The same sensors and workload should be used in both cases. This is necessary to determine the actual benefits in terms of traffic, auditing, reliability, and cost.

#### **4.5. Protocol Interoperability and Data Management**

The communication model must provide unambiguous event semantics regardless of the transport protocol. Each message contains a device identifier, parking-space identifier, event type, local sequence number, timestamp, schema version, and digital signature or authentication code. The eventId is generated so that one physical event cannot create two business operations. If the same message is received again, the Edge node returns an acknowledgement but does not change the state again. MQTT QoS 1 can therefore be used without creating duplicate charges or releasing the same space twice.

MQTT topics may be organized hierarchically by operator, facility, zone, controller, and parking space. Telemetry, commands, and acknowledgements are transmitted through separate channels. A controller may publish only to authorized topics. It subscribes only to commands addressed to its own actuating devices. Retained messages are not used for barrier commands. A Last Will message notifies the Edge layer when a device becomes unavailable. The sessions are protected by TLS and mutual certificate-based authentication. The broker access policy restricts both reading and publishing.

The payload may use JSON during the pilot implementation. Protocol Buffers or CBOR may be applied when system resources are limited. A version registry is required regardless of the selected format. A new version may add fields, but it must not change the meaning of previously published events. The Edge gateway verifies the required fields, permitted ranges, and signature before accepting a message. Invalid messages are transferred to a diagnostic queue. They are not forwarded to the smart contract. The blockchain layer is therefore not burdened with clearly corrupted or unauthorized data. The off-chain repository stores detailed telemetry, images, diagnostic files, and personal data for which a valid legal basis exists. A cryptographic hash is calculated for each significant document or evidential dataset. The blockchain stores the hash, document type, timestamp, and responsible organization.

During subsequent verification, the recalculated hash is compared with the stored value. This approach does not make the off-chain file publicly accessible. However, it makes it possible to determine whether the file has been modified. The retention period of the file is managed separately according to its purpose and the applicable rules.

#### **4.6. Security Privacy and Fault Tolerance**

The threat model includes falsified sensor events, cloned devices, replayed commands, compromised Edge nodes, unauthorized tariff changes, blockchain-node failures, and misuse of operator privileges. Preventive and detective measures are defined for each threat. The devices use unique cryptographic keys. Their certificates can be revoked. Administrative actions require an appropriate role and are recorded. A tariff version is approved before it becomes effective. Critical changes may require confirmation from two organizations. This limits the possibility of unilateral manipulation.

System availability is supported by a local cache of valid reservations and tariff rules. It is also supported by a persistent outbox and reconnection with progressively increasing intervals. The Edge node records an event before attempting external transmission. After a restart, unprocessed events are recovered according to their sequence numbers. If the blockchain gateway is unavailable, the local loop continues to monitor the physical state. New operations may be restricted according to the identified risk. However, emergency opening and safe exit remain independent of the blockchain connection. After communication is restored, the events are transmitted in their original order.

Privacy requires more than the pseudonymization of a licence-plate number. A repeated identifier may allow vehicle movements to be traced. The user identifier should therefore be replaced periodically or represented by a token with a limited purpose and validity period. The mobile application receives only the data required for the specific reservation. Analytical reports use aggregated values. Access to images is restricted and logged. Deleting personal off-chain data does not remove the corresponding blockchain event. However, only an irreversible hash without independent value for identifying the user remains in the ledger.

#### **4.7. Implementation Plan and Engineering Decision Criteria**

The pilot implementation consists of four stages. The first stage is a laboratory test bench with a sensor emulator, one controller, and a test blockchain network. The second stage is a closed area with real vehicles and parallel manual observation. The third stage is limited operation with users but without dynamic pricing. The fourth stage introduces multiple operators and integration with a payment service provider. Progression to the next

stage is permitted only when the criteria for accuracy, response time, recovery, and traceability have been satisfied. This approach reduces architectural risk before a large-scale investment is made.

The comparative experiment should use identical sensors, parking spaces, and scenarios for the centralized and Edge-blockchain configurations. The measured indicators include local latency, end-to-end latency, percentage of successfully processed events, network load, storage volume, and administrative time required to investigate a dispute. The blockchain configuration should not be evaluated only in terms of speed. Its expected benefits are improved verifiability and reduced dependence on a single operator. If these benefits are not demonstrated or the cost is disproportionate, the engineering solution should remain centralized.

A before-and-after analysis is required to evaluate the effect on traffic. The baseline period establishes occupancy, average search time, and the number of vehicle movements within the area. The same indicators are measured after implementation during comparable days and time intervals. The data are controlled for seasonal effects, temporary restrictions, and changes in overall demand. The effect of the information service is separated from the effect of the tariff. This makes it possible to determine whether the system actually reduces circulation or merely redistributes vehicles among nearby streets and parking facilities.

Quality management requires traceability from each requirement to its corresponding test. Every functional requirement receives a unique identifier. It is linked to an architectural component, smart contract function, and test scenario. A defect is recorded together with the relevant software version and system configuration. Retesting and regression testing are performed after the defect has been corrected. The indicators are reviewed periodically and not only during initial acceptance. This discipline is important for a system in which physical devices, machine recognition, payment operations, and the distributed ledger have different lifecycles.

The proposed model may also be applied to other limited urban resources. Such resources must have a measurable physical state, a temporary right of use, and a requirement for shared auditing. Examples include loading zones, charging stations, and short-term service areas. Transfer to another application is not automatic. A separate state machine, safe offline policy, tariff rules, and evidential events must be defined for each resource. The reusable principles include Edge decision-making, idempotent exchange, minimal on-chain recording, and separation of physical and contractual states.

The smart contract logic is also verified through invariants. A parking space cannot have two active reservations at the same time. A paid session may begin only after a valid access right or registered exception has

been established. Ending the session does not release the parking space until the physical model confirms that the vehicle is no longer present. The fee is calculated using the tariff version fixed at the beginning of the session. An administrative correction does not delete the original operation. These rules can be implemented as automated tests. They should be executed whenever the smart contract code is modified.

Load testing should distinguish between a continuous flow and short peaks. The continuous flow demonstrates the sustained capacity of the system. The peak test shows system behaviour during mass arrivals after an event or at the beginning of the working day. In addition to the mean value, queue lengths and upper latency percentiles are monitored. A system with acceptable average performance may still produce unacceptable occasional delays. Acceptance should therefore be based on a predefined maximum proportion of events exceeding the permitted threshold. It should not be based only on average performance.

The cost model covers the entire system lifecycle. Hardware is depreciated by equipment category because sensors, cameras, and the Edge platform have different service lives. The model includes spare devices, maintenance visits, operator training, cybersecurity, and software-version migration. Benefits are evaluated through fewer manual inspections, fewer disputes, better utilization of parking spaces, and shared services for multiple operators. The assessment therefore does not reduce economic efficiency to the cost of a single blockchain node.

The owner of each system function must be identified before implementation. The parking operator is responsible for the physical equipment. The technical administrator manages certificates and software versions. The municipal authority approves tariff rules where applicable. The payment service provider confirms the financial transaction without disclosing unnecessary payment data. The incident procedure defines who may pause the smart contract and who may authorize recovery. A clear allocation of responsibilities is a prerequisite for the practical value of the permissioned blockchain. It is not merely an organizational addition.

## 5. CONCLUSION

A conceptual Edge-blockchain architecture for intelligent parking has been developed. It integrates multimodal sensors, local controllers, NVIDIA Jetson Orin Nano, a permissioned blockchain network, a user application, and actuating devices. The main design principle is the separation of time-critical physical responses from asynchronous blockchain recording. This allows barriers, indicators, and local controllers to respond without waiting for blockchain confirmation. At the same time, significant events are recorded in a traceable and verifiable manner.

The scientific contribution lies in the coordinated modelling of the physical and contractual states of a parking space. The sensor vector, measurement normalization, occupancy estimation, and hysteresis switching have been formalized. The telemetry packet, hashing procedure, authenticated commands, and latency components have also been defined. Idempotent event processing is proposed. It limits the risk of repeated execution after communication interruptions or retransmissions. Only essential information is recorded on the blockchain. More detailed data can remain off-chain. This reduces system load, operating costs, and privacy risks.

The proposed architecture can be applied to municipal and private parking facilities managed by one or more operators. It supports the verifiable recording of occupancy, reservations, tariffs, payments, and control commands. This can facilitate the resolution of disputes between users, operators, and service providers. The modular structure allows gradual implementation. A limited pilot area can be developed first. The system can then be expanded by adding new parking spaces, Edge nodes, and organizations to the permissioned network.

The study is conceptual. No results from a physical prototype or real-world operation are presented at this stage. The next step is to develop a test bench and a pilot installation. Occupancy-detection accuracy, local response time, blockchain latency, throughput, and resource consumption should be measured. The centralized and proposed Edge-blockchain implementations should be compared under identical scenarios. Operating costs should also be evaluated for different numbers of parking spaces and transactions. These tests will identify the limits of applicability and provide a quantitative basis for deployment in a real urban environment.

## References:

- Aboudolas, K., & Geroliminis, N. (2013). Perimeter and boundary flow control in multi-reservoir heterogeneous networks. *Transportation Research Part B: Methodological*, 55, 265–281. <https://doi.org/10.1016/j.trb.2013.07.003>
- Ahmed, S., Soaibuzzaman, Rahman, M. S., & Rahaman, M. S. (2019). A blockchain-based architecture for integrated smart parking systems. *2019 IEEE International Conference on Pervasive Computing and Communications Workshops*, 177–182. <https://doi.org/10.1109/PERCOMW.2019.8730772>

- Al Amiri, W., Baza, M., Banawan, K., Mahmoud, M., Alasmay, W., & Akkaya, K. (2020). Towards secure smart parking system using blockchain technology. *2020 IEEE 17th Annual Consumer Communications & Networking Conference*, 1–6. <https://doi.org/10.1109/CCNC46108.2020.9045674>
- Arnott, R., de Palma, A., & Lindsey, R. (1993). A structural model of peak-period congestion: A traffic bottleneck with elastic demand. *American Economic Review*, 83(1), 161–179.
- Daganzo, C. F. (2007). Urban gridlock: Macroscopic modeling and mitigation approaches. *Transportation Research Part B: Methodological*, 41(1), 49–62. <https://doi.org/10.1016/j.trb.2006.03.001>
- Fahim, A., Hasan, M., & Chowdhury, M. A. (2021). Smart parking systems: Comprehensive review based on various aspects. *Heliyon*, 7(5), e07050. <https://doi.org/10.1016/j.heliyon.2021.e07050>
- Geroliminis, N. (2015). Cruising-for-parking in congested cities with an MFD representation. *Economics of Transportation*, 4(3), 156–165. <https://doi.org/10.1016/j.ecotra.2015.04.001>
- Haddad, J., Ramezani, M., & Geroliminis, N. (2013). Cooperative traffic control of a mixed network with two urban regions and a freeway. *Transportation Research Part B: Methodological*, 54, 17–36. <https://doi.org/10.1016/j.trb.2013.04.010>
- Hagman, O. (2006). Morning queues and parking problems: On the broken promises of the automobile. *Mobilities*, 1(1), 63–74. <https://doi.org/10.1080/17450100500489289>
- Hyperledger Foundation. (n.d.). *Hyperledger Fabric model*. [https://hyperledger-fabric.readthedocs.io/en/latest/fabric\\_model.html](https://hyperledger-fabric.readthedocs.io/en/latest/fabric_model.html)
- Imbugwa, G., Mazzara, M., & Distefano, S. (2020). Developing a mobile application using open source parking management system on Ethereum smart contracts. *Journal of Physics: Conference Series*, 1694, Article 012022. <https://doi.org/10.1088/1742-6596/1694/1/012022>
- Jennath, H. S., Adarsh, S., Chandran, N. V., Ananthan, R., Sabir, A., & Asharaf, S. (2019). Parkchain: A blockchain powered parking solution for smart cities. *Frontiers in Blockchain*, 2, Article 6. <https://doi.org/10.3389/fbloc.2019.00006>
- Kalbhor, A., Nair, R. S., Phansalkar, S., Sonkamble, R., Sharma, A., Mohan, H., Wong, C. H., & Lim, W. H. (2024). PARKTag: An AI-blockchain integrated solution for an efficient, trusted, and scalable parking management system. *Technologies*, 12(9), Article 155. <https://doi.org/10.3390/technologies12090155>
- Keyvan-Ekbatani, M., Kouvelas, A., Papamichail, I., & Papageorgiou, M. (2012). Exploiting the fundamental diagram of urban networks for feedback-based gating. *Transportation Research Part B: Methodological*, 46(10), 1393–1403. <https://doi.org/10.1016/j.trb.2012.06.008>
- Khan, S. N., Loukil, F., Ghedira-Guegan, C., Benkhelifa, E., & Bani-Hani, A. (2021). Blockchain smart contracts: Applications, challenges, and future trends. *Peer-to-Peer Networking and Applications*, 14, 2901–2925. <https://doi.org/10.1007/s12083-021-01127-0>
- NVIDIA Corporation. (n.d.). *Jetson Orin Nano Developer Kit user guide*. <https://docs.nvidia.com/jetson/orin-nano-devkit/user-guide/latest/>
- OASIS. (2019). *MQTT version 5.0*. <https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html>
- Shoup, D. C. (1997). The high cost of free parking. *Journal of Planning Education and Research*, 17(1), 3–20. <https://doi.org/10.1177/0739456X9701700102>
- Vickrey, W. S. (1969). Congestion theory and transport investment. *American Economic Review*, 59(2), 251–260.
- Zakharenko, R. (2019). The economics of parking occupancy sensors. *Economics of Transportation*, 17, 14–23. <https://doi.org/10.1016/j.ecotra.2018.11.001>
- Zhang, C., Zhu, L., & Xu, C. (2023). BSDP: Blockchain-based smart parking for digital-twin empowered vehicular sensing networks with privacy protection. *IEEE Transactions on Industrial Informatics*, 19(5), 7237–7246. <https://doi.org/10.1109/TII.2022.3223211>

---

**Raya Stoyanova**

Department of Mechanical Engineering,  
Faculty of Engineering, South-West  
University 'Neofit Rilski'-  
Blagoevgrad, Bulgaria  
City, Blagoevgrad  
Country Bulgaria  
[raikach@abv.bg](mailto:raikach@abv.bg)  
ORCID 0000-0001-9633-4467

**Filip Batalov**

Department of Mechanical Engineering,  
Faculty of Engineering, South-West  
University 'Neofit Rilski'-  
Blagoevgrad, Bulgaria  
City, Blagoevgrad  
Country Bulgaria  
[tangra@mail.ru](mailto:tangra@mail.ru)  
ORCID 0000-0001-7328-1526

---

