systems

MDPI

Article

From Agent-Based Simulation to an IoT-Enabled Prototype for Infrastructure-Less Cooperative Parking Guidance †

Davide De Soricellis, Riccardo Todaro, Filippo Muzzini * and Angelo Ferrando

Department of Physics, Informatics and Mathematics, University of Modena and Reggio Emilia, 41125 Modena, Italy; 309004@studenti.unimore.it (D.D.S.); 260485@studenti.unimore.it (R.T.); angelo.ferrando@unimore.it (A.F.)

* Correspondence: filippo.muzzini@unimore.it

† This paper is an extended version of our paper Evaluating Infrastructure-Less Cooperative Parking Guidance via Agent-Based Simulation published in Proceedings of the PAAMS 2025, Lille, France, 25–27 June 2025.

Highlights

Please indicate how your work links to systems science via your contributions to systems practice, theory, and/or methodology.

- An infrastructure-less parking-evidence acquisition and heatmap-visualisation pipeline was implemented as a proof-of-concept prototype.
- A score-based recommendation function was developed to match parking areas with user preferences.

What are the main findings and/or the implications of the main findings?

- The prototype demonstrates the feasibility of a low-cost implementation pathway that does not require dedicated roadside parking infrastructure.
- Simulation results indicate that preference-aware cooperative guidance can reduce parking-search time under the evaluated conditions.

Abstract

Urban parking search remains a major source of congestion, travel delay, and unnecessary emissions, especially in areas where free on-street parking is not instrumented by dedicated infrastructure. This article investigates an infrastructure-less cooperative parking-guidance approach in which participating vehicles contribute lightweight parking-related evidence that is aggregated into shared heatmaps. The proposed model formalises parking-event semantics, heatmap representation, evidence updates, and a score-based parking-selection strategy that balances parking opportunity against walking distance. A SUMO-based simulation campaign evaluates the model under different user-preference parameters, system-adoption rates, and levels of initially available information. The results show that shared parking evidence can reduce search time under the evaluated conditions, including scenarios with partial adoption, while also highlighting the influence of user preferences and cold-start information availability on system performance. To examine the implementation feasibility of the information-generation mechanism assumed by the model, the article additionally presents a compact proof-of-concept pipeline based on a Bluetooth Low Energy beacon, a smartphone application, and a lightweight back-end service. Functional testing demonstrates that parking-related state-change events can be generated, transmitted, stored, spatially aggregated, and visualised without dedicated roadside parking sensors. This prototype serves only as an implementation-feasibility demonstration and does not constitute field validation of the guidance strategy or establish real-world accuracy, latency, reliability, or deployment readiness.

Check for updates

Academic Editor: Zhou He

Received: 26 May 2026
Revised: 1 September 2026
Accepted: 2 September 2026
Published: 4 September 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

Keywords: smart parking; multi-agent systems; SUMO; cooperative guidance; Bluetooth Low Energy; mobile application; heatmap-based guidance; Internet of Things

1. Introduction

Finding parking in urban areas remains a pervasive and costly source of inefficiency. Drivers often spend a non-negligible amount of time cruising near their destinations, producing additional traffic, increased travel uncertainty, unnecessary fuel consumption, and higher emissions [1,2]. The problem is particularly acute for free on-street parking and other open parking scenarios in which occupancy changes rapidly and real-time information is rarely available [3,4]. In such settings, conventional smart-parking approaches based on dedicated sensors, cameras, barriers, or smart meters are difficult to deploy at scale, especially when the parking resource is geographically dispersed and not directly managed by a single operator [5,6].

This tension between the practical importance of parking information and the cost of the infrastructure required to collect it motivates infrastructure-light or infrastructure-less alternatives. The core intuition of the approach presented in this article is that vehicles do not necessarily need fixed roadside sensing to cooperate. If participating cars can contribute lightweight evidence about the occupancy state of nearby parking areas, then the community can progressively build a useful representation of parking opportunity.

The proposed system is built around parking-related events rather than direct slot-level sensing. A participating vehicle contributes evidence when its parking state changes, and this evidence is aggregated into a shared heatmap that represents the relative opportunity of different urban areas. Cooperative agents can then use this representation to choose parking areas that balance the likelihood of finding a space against the walking distance to the destination.

This article also addresses the practical question of how such evidence can be generated without dedicated roadside hardware. We present an IoT-enabled prototype in which a Bluetooth Low Energy (BLE) beacon placed in the vehicle interacts with a smartphone application carried by the driver. Beacon connection and disconnection events act as a lightweight proxy for whether the driver is inside the vehicle or has moved away from it. Combined with geolocation, these events provide the raw data required to build parking heatmaps without fixed sensors, cameras, barriers, or smart meters.

The primary focus of this article is the evaluation of an infrastructure-less cooperative parking-guidance model under different behavioural and information-sharing conditions. The simulation study provides the quantitative evidence used to investigate the effects of user preferences, system adoption, and initial information availability. To complement this evaluation, we also present a proof-of-concept implementation of the parking-evidence acquisition pipeline assumed by the model. The prototype is intended to demonstrate the practical feasibility of generating and aggregating such evidence without dedicated roadside sensing infrastructure; it is not used as an additional experimental validation of the guidance strategy.

The main contributions of this article are the following:

- It consolidates, extends, and formalises the cooperative parking-guidance model first introduced in [7], including the event semantics, heatmap abstraction, update rules, and score-based parking-selection strategy;
- It provides a systematic SUMO-based evaluation of the model, investigating how parking-search performance is affected by user preferences, cooperative-system adoption, and the amount of information available at the beginning of the search process;

- It presents a proof-of-concept implementation of the data-acquisition and aggregation mechanism assumed by the model, showing how parking-related state-change evidence can be generated using a BLE beacon and smartphone, transmitted to a lightweight back-end service, and converted into a shared spatial representation without dedicated roadside sensors;
- It discusses the relationship between the simulation assumptions and their practical implementation, while explicitly identifying the limitations of the current prototype and the additional validation required before real-world deployment.

Table 1 clarifies the distinct roles of the simulation study and the proof-of-concept implementation, indicating the type of evidence used to support each objective of the article.

Table 1. Study objectives, supporting evidence, and corresponding claims.

Study Objective	Supporting Evidence	Claim Supported
RQ1: user-preference effects	SUMO-based simulation	Quantifies how the opportunity–distance trade-off affects parking-search performance.
RQ2: system-adoption effects	SUMO-based simulation	Quantifies how the proportion of participating vehicles influences parking-search performance under different information conditions.
RQ3: initial-information effects	SUMO-based simulation	Quantifies how the availability of pre-existing heatmap information affects the initial phases of cooperative parking search.
Implementation-feasibility objective	Proof-of-concept prototype and functional testing	Demonstrates that parking-related state-change evidence can be generated, transmitted, stored, spatially aggregated, and visualised using commodity components without dedicated roadside parking sensors.
Deployment implications	Interpretation of simulation and prototype limitations	Identifies the assumptions, validation gaps, and engineering steps that must be addressed before real-world deployment.

To evaluate the behaviour of the proposed cooperative parking-guidance model, this work addresses three main research questions:

- RQ1. How do individual user preferences (balancing search time against walking distance to the final destination) impact overall parking performance?
- RQ2. How does the system's market penetration (i.e., the proportion of vehicles utilising the shared heatmap) influence overall efficiency under varying information conditions?
- RQ3. What is the impact of pre-existing information on system efficacy during the initial phases of the search process?

These research questions concern the behavioural performance of the cooperative guidance model and are therefore investigated through the simulation study. In addition to these research questions, the paper pursues a complementary implementation-feasibility objective. Specifically, it examines whether the parking-related evidence assumed by the model can be generated, transmitted, stored, aggregated, and visualised without dedicated roadside sensing infrastructure. The IoT prototype addresses this objective by instantiating the corresponding data-acquisition pipeline using a BLE beacon, a smartphone application, and a lightweight back-end service. Accordingly, the prototype is not used to answer RQ1–RQ3 and is not presented as a field validation of the guidance strategy; its role is to demonstrate, at proof-of-concept level, that the information-generation mechanism assumed by the model can be implemented using commodity hardware and software components.

The remainder of the paper is organised as follows. Section 2 reviews the relevant literature and positions the proposed approach. Section 3 formalises the cooperative parking-guidance model. Section 4 presents the simulation campaign used to address RQ1–RQ3 and analyses the effects of user preferences, system adoption, and initial information availability. Section 5 presents a condensed proof-of-concept realisation of the parking-evidence acquisition and aggregation pipeline assumed by the model. Section 7 relates the simulation findings and implementation feasibility to the requirements and limitations of real-world deployment. Finally, Section 8 concludes the paper.

2. Related Work and Positioning

The smart-parking literature is broad and heterogeneous. Existing work spans sensor-rich occupancy monitoring, reservation and allocation systems, parking prediction, mobile crowdsensing, and agent-based coordination. For the purposes of this article, the most relevant distinction is not merely between centralised and decentralised systems, but among approaches that presuppose dedicated parking infrastructure, approaches that exploit already-available mobile or vehicular sensing capabilities, and approaches that rely on standardised vehicle-to-everything communication. These families make different trade-offs in terms of sensing accuracy, deployment cost, communication latency, equipment penetration, user involvement, and suitability for open on-street parking.

2.1. Infrastructure-Centric Smart Parking

A large portion of smart-parking research relies on dedicated occupancy-sensing infrastructure. Surveys consistently report the prevalence of embedded sensors, computer-vision systems, smart meters, barrier-based parking facilities, and centralised services that monitor or reserve parking resources [3–5]. These approaches can provide accurate and regularly updated information and are particularly well suited to structured parking facilities or highly managed curbside areas. In open urban environments, however, especially where free parking is dispersed across ordinary streets, the deployment and maintenance costs of pervasive infrastructure become a major limitation [6].

Another important research direction focuses on forecasting. Reviews such as [8] show that prediction-based systems increasingly use machine learning to anticipate parking demand and occupancy. Prediction is highly valuable, but it is not a substitute for scalable data collection. A predictive system still requires historical observations, a real-time sensing pipeline, or both. The data-acquisition problem therefore remains central even when the final objective is predictive analytics rather than direct occupancy monitoring.

2.2. Agent-Based and Cooperative Parking Guidance

Agent-based and multi-agent approaches are also well represented in the literature. Some works model parking as a strategic or allocation problem, for instance through assignment games or distributed reservation mechanisms [9–11]. Others propose architectures where agents represent vehicles, users, parking operators, or prices, often with explicit coordination or negotiation protocols [12–15]. Parking guidance has also been studied through reactive multi-agent systems and optimisation-inspired approaches such as ant-colony-based methods [16,17]. More recent work considers reinforcement learning and partially connected vehicles [18].

These contributions confirm that parking is a natural application domain for agent-based reasoning. However, many of them assume reservable parking spaces, centrally managed resources, rich communication support, or occupancy information produced by an external sensing infrastructure. Our setting is intentionally different: we consider open, non-reservable, infrastructure-poor urban parking, where useful guidance must emerge

from partial, noisy, and incrementally collected evidence. Cooperation in our model does not imply that every parking space is known or controllable; it means that participating users contribute observations from which a shared representation of parking opportunity can progressively be constructed.

2.3. Smartphone-Centric and Mobile-Crowdsensing Approaches

Smartphones have long been investigated as an alternative to fixed parking sensors because they integrate location, inertial sensing, wireless communication, storage, and user-interaction capabilities. Early systems demonstrated that parking-related events could be inferred from changes in a driver's mobility or wireless context rather than from sensors installed in individual parking spaces.

PhonePark detects parking and unparking activities from transportation-mode transitions observed through mobile-phone sensors and also considers the establishment or loss of the Bluetooth connection between the phone and the vehicle as a parking-related cue [19]. ParkSense instead uses urban Wi-Fi observations: it matches Wi-Fi signatures to detect that the driver has returned to the parked vehicle and analyses changes in the observed access points to infer that the vehicle has begun moving [20]. PocketParker adopts a crowdsourcing model in which smartphone observations of arrivals and departures are aggregated to estimate parking-lot availability without instrumenting individual spaces [21]. These systems established the feasibility of infrastructure-light parking sensing, while also exposing recurring challenges concerning sensor ambiguity, energy consumption, background execution, location accuracy, and the number of participating users.

Recent work continues this line of research while addressing data governance and model construction. FedPark, for example, combines smartphone-based crowdsensing with federated learning so that parking-availability models can be trained without centralising all raw observations from participating drivers [22]. More generally, contemporary IoT architectures combine mobile devices, vehicle-side electronics, cloud services, and edge components to separate data acquisition from storage, aggregation, and decision support [23].

Our approach belongs to this smartphone-centric family, and we do not claim to be the first system to use Bluetooth information as a cue for parking-state detection. The closest conceptual precedent is PhonePark. The distinction lies in the role assigned to the Bluetooth connection and in its integration with the guidance model. We use a dedicated low-cost BLE beacon installed in the vehicle as a simple user–vehicle proximity source, independently of an integrated vehicle infotainment system. Persistent connection and disconnection transitions generate timestamped and geolocated evidence that is transmitted automatically to a back end. These observations are then aggregated at the area level and connected to a cooperative, agent-based parking-guidance strategy.

The proposed system therefore does not attempt to infer the exact state of every parking space from a complete smartphone trajectory. Nor does it require the driver to report a free or occupied space manually. It uses automatically generated state-transition events as uncertain evidence from which a shared heatmap of parking opportunity is constructed. This choice reduces the sensing and interaction burden, at the cost of providing indirect and potentially noisy evidence rather than direct slot-level occupancy.

2.4. V2X-Based Cooperative Parking and Communication Trade-Offs

A separate family of cooperative parking systems relies on vehicle-to-everything communication. IEEE 802.11p-derived DSRC/ITS-G5 and cellular V2X technologies support direct or infrastructure-mediated exchange among vehicles, roadside units, central services, and other road users [24]. In a parking context, such communication can be used to distribute occupancy information, advertise parking facilities, coordinate reservations,

or guide vehicles towards available areas. Recent work has proposed end-to-end parking guidance in which C-V2X provides wider-area guidance while a DSRC vehicular network supports local guidance within or near a parking facility [25].

Parking information is also explicitly represented in current cooperative-ITS standardisation. The ETSI Parking Availability Service defines a standard Point-of-Interest Message for Parking Availability, or POIM-PA, through which static and dynamic parking information can be broadcast locally or delivered to selected vehicles. The specification permits different communication profiles, including ITS-G5, LTE-V2X, and long-range cellular access [26]. V2X-based parking support can therefore benefit from standardised message semantics, direct local dissemination, integration with vehicle navigation systems, and communication paths designed for vehicular environments.

These benefits are accompanied by a different deployment profile from the one considered in this article. Direct V2X communication requires compatible on-board units in participating vehicles. Infrastructure-assisted configurations may additionally require roadside units, parking-side ITS stations, security credentials, spectrum and channel management, and integration with municipal or commercial parking-management systems. Their effectiveness consequently depends on vehicle-equipment penetration and, in some configurations, on the geographic coverage and maintenance of roadside infrastructure.

The proposed architecture avoids these requirements by using commodity mobile connectivity and a low-cost in-vehicle beacon. Importantly, BLE is not used as a vehicle-to-vehicle or vehicle-to-infrastructure communication technology. It is used only over a very short range to infer changes in the relationship between one driver and their own vehicle. The resulting event is then transmitted by the smartphone to a cloud-based service using conventional Internet connectivity. The comparison is therefore not between BLE and C-V2X as alternative radio technologies for the same communication task; it is between two different system architectures.

The smartphone–cloud solution offers a lower entry barrier and can be used by legacy vehicles without a V2X on-board unit. It is also well matched to city-scale evidence aggregation and to parking information, whose timing requirements are generally less stringent than those of collision avoidance or cooperative manoeuvring. Conversely, it depends on mobile-network availability, cloud-service reachability, smartphone operating-system policies, and application adoption. Its communication delay is less predictable than direct local V2X dissemination, and its parking-state observations remain indirect.

The two paradigms are not mutually exclusive. A future hybrid architecture could accept events from smartphone participants, V2X-equipped vehicles, and parking-side infrastructure. The confidence parameter introduced in our evidence model provides a natural mechanism for assigning different weights to events according to their source, sensing method, freshness, and communication provenance.

2.5. Scientific Gaps in Existing Research

Beyond the technological differences among sensing and communication architectures, the literature reviewed above reveals several scientific questions that remain insufficiently connected in the context of open, non-reservable urban parking.

First, parking-information acquisition and parking guidance are frequently treated as distinct problems. Infrastructure-centric and mobile-crowdsensing research primarily investigates how parking occupancy or parking-related events can be observed, inferred, or predicted, whereas agent-based, allocation, and reservation approaches commonly reason over parking information that is assumed to be already available. This separation leaves an important question: whether indirect, partial, and incrementally collected user-generated evidence can remain sufficiently informative to support effective collective

guidance. In open parking environments, this is not merely a sensing-accuracy problem, because uncertainty in the observations propagates directly into the decisions made by participating drivers.

Second, participation in a crowdsourced parking system is not only a deployment requirement but also a factor affecting the dynamics of the information itself. When only a fraction of vehicles contributes observations, the spatial coverage and freshness of the shared representation depend on market penetration. At the same time, participating vehicles consume that same representation to make parking decisions. The resulting relationship between partial adoption, information quality, and system-level parking performance therefore deserves explicit investigation rather than being treated solely as a limitation of crowdsensing.

Third, infrastructure-less cooperative systems face a knowledge-formation problem during their initial operation. When little or no historical evidence is available, the shared representation must progressively emerge from new observations. The effect of this cold-start condition on parking-search performance, and the extent to which pre-existing information accelerates the usefulness of cooperative guidance, constitute a distinct research question.

Finally, the effect of heterogeneous user objectives remains relevant when guidance is based on incomplete evidence. Drivers may value proximity to their destination differently from a reduction in parking-search time. Consequently, it is necessary to understand how this preference trade-off interacts with cooperative information rather than considering only a single system-wide optimisation objective.

These gaps motivate the scientific questions investigated in this work. RQ1 studies the interaction between user preferences and parking-search performance; RQ2 examines the effect of partial system penetration on the benefits of cooperative information sharing; and RQ3 investigates the role of pre-existing information and the associated cold-start problem. The proposed model-to-prototype pipeline additionally connects the acquisition of parking-related evidence with the decision-support model that consumes it. Real-world evidence accuracy, communication latency, behavioural heterogeneity, and longitudinal deployment effects remain outside the claims of the present study and constitute important directions for subsequent empirical validation.

2.6. Positioning of This Work

The scientific positioning of this work follows from the gaps identified above. Rather than proposing another parking-sensing technology in isolation, we study the behaviour of a cooperative guidance process when the information on which decisions depend is partial, incrementally generated by users, and therefore sensitive to adoption and initial knowledge. The IoT prototype complements this investigation by demonstrating a feasible mechanism through which the evidence assumed by the model can be generated in practice. Within this scientific framing, the contribution integrates:

- An event-based model of partial parking evidence;
- An agent-based guidance strategy that operates under partial adoption;
- An automated BLE-based mechanism for generating user–vehicle proximity transitions;
- A smartphone–cloud pipeline for storing and aggregating those transitions;
- A shared, area-level heatmap that connects the prototype data acquisition layer to the simulation model.

Our goal is not to compete with high-accuracy infrastructure-heavy parking management in controlled facilities, or with low-latency V2X services in fully connected transport environments. Instead, we target open urban parking, where spaces are not reservable,

fixed occupancy information is scarce, legacy vehicles remain prevalent, and the economic threshold for deployment is low.

Table 2 summarises this positioning. It should not be read as claiming that one family of solutions is universally superior to another. Instead, it identifies the contexts in which the different acquisition and communication architectures are most appropriate, and the limitations that remain when they are applied to open free parking.

Table 2. Positioning of the proposed approach within the smart-parking landscape.

Approach Family	Parking-Information Source	Communication Path	Main Deployment Advantage	Typical Limitation in Open Free Parking
Embedded sensing/cameras	Fixed roadside or parking-side sensors	Local infrastructure and central service	Accurate and regularly updated occupancy information	Installation, calibration, coverage, and maintenance costs over large urban areas
Reservation/allocation systems	Managed inventory of reservable spaces	Operator platform and user application	Strong coordination and guaranteed allocation when resources are controlled	Not directly applicable to ordinary, non-reservable on-street spaces
Prediction-oriented systems	Historical and/or real-time occupancy traces	Centralised or cloud-based analytics	Can anticipate demand and compensate for incomplete instantaneous observations	Prediction quality depends on the availability, representativeness, and freshness of the underlying data
Smartphone and mobile crowdsensing	GPS, inertial, Wi-Fi, Bluetooth, mobility transitions, or user reports	Mobile Internet and cloud service	Uses commodity devices and avoids fixed sensing at every parking space	Sensor ambiguity, battery and background-execution constraints, participation, and potentially uneven coverage
DSRC/ITS-G5 or C-V2X cooperative parking	Vehicle, roadside, parking-facility, or central ITS observations	Direct V2V/V2I communication and/or cellular V2X	Low-latency local dissemination, standardised messages, and integration with connected vehicles	Requires compatible on-board equipment and may require roadside units, security infrastructure, and operator integration
Infrastructure-less cooperative guidance (this work)	BLE user–vehicle proximity transitions combined with smartphone geolocation	Short-range BLE for local detection; conventional mobile Internet for event dissemination	Low entry cost, support for legacy vehicles, automated reporting, and area-level cooperative aggregation	Data quality depends on adoption, proxy reliability, mobile connectivity, and event freshness

The remainder of the paper develops this integrated perspective. We first formalise how parking-related events are transformed into shared evidence and used by the cooperative guidance strategy. We then show how the same event semantics can be realised through the BLE-enabled mobile prototype.

3. System Model: Cooperative Guidance Without Roadside Infrastructure

This section formalises the system model underlying the proposed cooperative parking-guidance approach. We start from the problem setting, then define the event semantics, the heatmap representation, and the scoring function used to guide parking search.

3.1. Problem Setting and Core Assumptions

We consider an urban area in which parking is mostly represented by ordinary free on-street spaces or other open-access spots that cannot be reserved in advance. Drivers wish to reach a destination area, but they may be willing to trade off a short final walking distance against a shorter search process. Vehicles are heterogeneous in the sense that some participate in the cooperative system and others do not. Participating vehicles can

receive aggregated parking evidence and, crucially, can also contribute evidence when their parking state changes.

The modelling assumptions are deliberately lightweight. We do not assume smart parking meters, curbside sensors, cameras, or dedicated vehicular communication infrastructure. Instead, we assume that a participating user carries a smartphone and that the corresponding vehicle contains a simple BLE beacon. The beacon itself is not a parking sensor. Rather, its connection status acts as a proxy for whether the driver is inside the vehicle or has moved away from it. This is sufficient to generate two kinds of parking-related evidence: a vehicle that has just become parked, and a vehicle that is likely to be leaving a parking position.

The model is therefore designed around events rather than around direct ground-truth occupancy sensing. This is important both conceptually and practically. Conceptually, it reflects the fact that in open urban parking we often have only partial evidence. Practically, it enables a very cheap implementation path, because event generation can be supported by commodity devices.

3.2. Event Semantics and State Changes

The prototype and the model share the same two application-level states:

- In motion: The smartphone is connected to the in-vehicle beacon, which is interpreted as evidence that the driver is inside the vehicle and the car is active or about to move.
- Parked: The smartphone is no longer connected to the beacon, which is interpreted as evidence that the user has moved away from the vehicle and the car is occupying a spot.

Figure 1 summarises both the conceptual state model and the transition semantics used by the application. The initial state is determined when the application starts, but only subsequent state changes are transmitted to the server. This design prevents the system from reporting stale or ambiguous information at launch time and focuses the data pipeline on meaningful parking-related transitions.

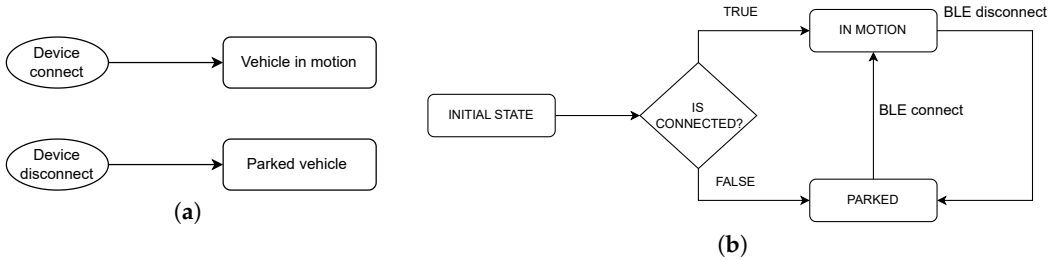


Figure 1. State semantics underlying the infrastructure-less parking evidence pipeline. (**a**) Conceptual user/vehicle states. (**b**) Application-level state transitions used to generate parking evidence.

A parking-related event can therefore be described abstractly as a tuple

$$e = \langle \sigma, \ell, t \rangle, \tag{1}$$

where $\sigma \in \{\texttt{parked}, \texttt{departing}\}$ is the event type, ℓ is the geolocated position associated with the transition, and t is the timestamp. In the current prototype, the raw event store preserves the state label explicitly. In the simulation model, the same information is abstracted into heatmap updates used for decision support.

This event-based interpretation is particularly useful because it decouples evidence generation from decision making. A city or community can first deploy the event-collection layer. Only later, once enough data have accumulated, must the system commit to a specific recommendation policy.

3.3. Heatmap Representation

The conference paper [7] introduced the notion of a shared heatmap to summarise parking opportunity. We retain that idea here and explain it in a more systematic way. The map of the urban area is discretised into a grid of cells, each representing a small region in which parking activity may occur. Let

$$H_{map} = \{h_{i,j} \mid 1 \leq i \leq M,\ 1 \leq j \leq N\} \tag{2}$$

be the heatmap over M rows and N columns. Each cell stores a value that summarises the parking evidence currently associated with that area.

Figure 2 shows the conceptual grid used by the proposed model. The purpose of the heatmap is not to represent each physical parking slot exactly. Instead, it provides a mesoscopic view of parking opportunity by aggregating evidence spatially. This is appropriate for routing and search, because drivers typically reason in terms of promising areas rather than individual curbside cells.

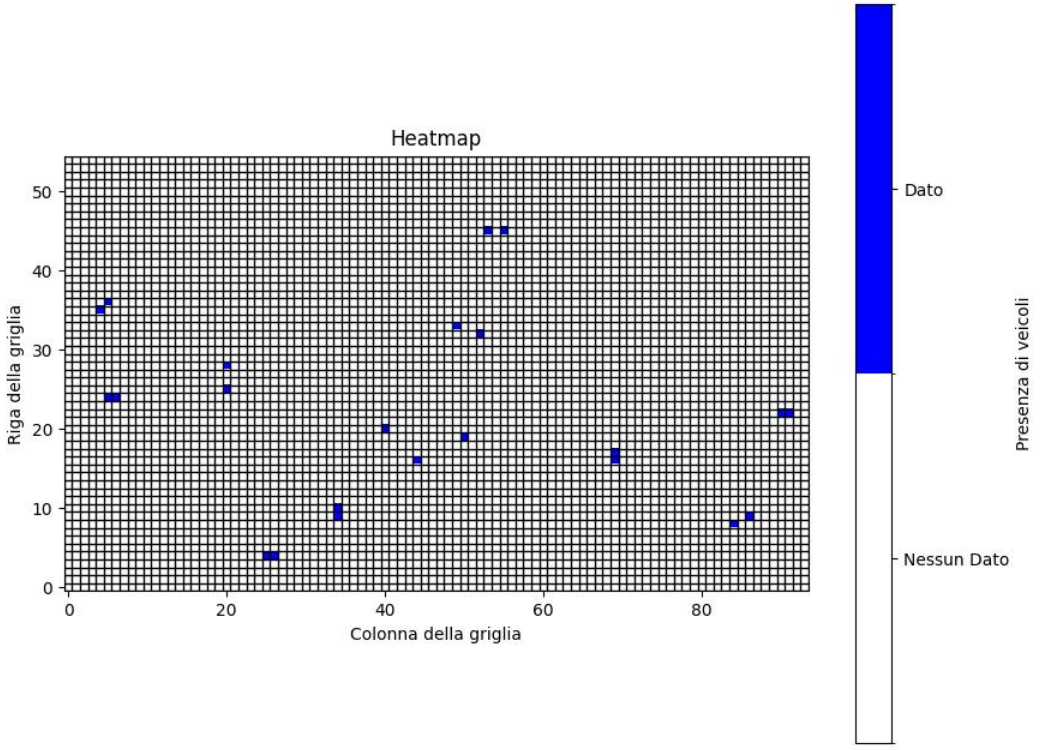


Figure 2. Conceptual heatmap representation used by the cooperative parking model. Each cell aggregates local parking-related evidence over an area rather than representing a single physical slot.

In the revised formulation, the heatmap is no longer treated as a simple integer counter. Each cell stores a confidence-weighted estimate of parking opportunity derived from two evidence layers: release evidence and occupation evidence. Release evidence corresponds to events suggesting that a parking space has become, or is about to become, available; occupation evidence corresponds to events suggesting that a vehicle has occupied a space. Both evidence layers are spatially propagated to nearby cells and temporally discounted so that old observations progressively lose influence. This makes the heatmap an operational estimate of parking opportunity rather than a raw count of events.

For clarity, Table 3 summarises the main notation used in the rest of the section.

Table 3. Main notation for the cooperative parking model.

Symbol	Meaning
H_{map}	Shared parking heatmap over the urban area
$h_{i,j}^{(t)}$	Normalised parking-opportunity value associated with cell (i,j) at time t
$R_{i,j}^{(t)}$	Confidence-weighted release evidence associated with cell (i,j) at time t
$O_{i,j}^{(t)}$	Confidence-weighted occupation evidence associated with cell (i,j) at time t
c_e	Confidence associated with event e, with $c_e \in [0,1]$
$K_\rho(\cdot)$	Spatial kernel controlling how strongly an event affects neighbouring cells
λ	Temporal decay rate controlling how quickly old evidence loses influence
$q_{i,j}^{(t)}$	Confidence/coverage factor associated with the evidence available for cell (i,j)
$\langle T_i, T_j \rangle$	Cell corresponding to the driver's destination area
$H_{score}(i,j)$	Estimated parking opportunity in cell (i,j)
$D(\langle i,j \rangle, d, \langle T_i, T_j \rangle)$	Normalised distance factor between candidate cell and destination
$S(i,j)_T$	Final score used to rank cell (i,j) for destination T
α	Parameter balancing parking opportunity and walking distance
d	Radius delimiting the local region of interest around the destination

3.4. Score-Based Parking Selection

The role of the heatmap is to support parking decisions. A cooperative vehicle does not simply ask "where is the closest spot?"; it asks which nearby area offers the best compromise between being promising and being close enough to the destination. This compromise is encoded through the following scoring function:

$$S(i,j)_T^{(t)} = \alpha \cdot H_{score}(i,j,t) + (1-\alpha) \cdot D(\langle i,j \rangle, d, \langle T_i, T_j \rangle). \tag{3}$$

where $\alpha \in [0,1]$ controls the trade-off between parking opportunity and destination proximity. Both terms in Equation (3) are dimensionless and normalised to the interval $[0,1]$. Therefore, the score is a convex combination of two comparable quantities: $H_{score}(i,j,t)$ represents the normalised parking-opportunity component, while $D(\cdot)$ represents the normalised proximity component.

The availability term is derived from the heatmap evidence rather than assumed directly. Let $R_{i,j}^{(t)}$ denote the release evidence accumulated for cell (i,j) at time t, and let $O_{i,j}^{(t)}$ denote the corresponding occupation evidence. We first compute a smoothed estimate of the probability that the cell contains available parking opportunity:

$$\hat{p}_{free}(i,j,t) = \frac{\varepsilon + R_{i,j}^{(t)}}{2\varepsilon + R_{i,j}^{(t)} + O_{i,j}^{(t)}}, \tag{4}$$

where $\varepsilon > 0$ is a small smoothing constant. This term compares evidence in favour of release against evidence in favour of occupation.

Because cells with little or no evidence should not be treated as highly reliable, we also define a confidence/coverage factor:

$$q_{i,j}^{(t)} = 1 - e^{-\left(R_{i,j}^{(t)} + O_{i,j}^{(t)}\right)}. \tag{5}$$

The final heatmap-based opportunity score is then

$$H_{score}(i,j,t) = q_{i,j}^{(t)} \cdot \hat{p}_{free}(i,j,t). \tag{6}$$

Thus, a cell receives a high score only when recent and reliable evidence suggests availability, while cells with insufficient evidence remain weakly attractive. If an explicit capacity estimate $P_{total}(i,j)$ is available, the corresponding occupancy estimate can be written as

$$\hat{P}_{occupied}(i,j,t) = P_{total}(i,j)\Big(1 - \hat{p}_{free}(i,j,t)\Big), \tag{7}$$

which makes explicit the relationship between the heatmap evidence and the occupancy estimate used by the guidance policy.

The distance term is computed as

$$D(\langle i,j\rangle, d, \langle T_i, T_j\rangle) = \max\left(0, 1 - \frac{E_{dist}(\langle i,j\rangle, \langle T_i, T_j\rangle)}{d}\right), \tag{8}$$

where E_{dist} is the Euclidean distance between the candidate cell and the destination cell, and d is the search radius used to delimit the relevant local neighbourhood. This definition maps the distance component into a normalised proximity score: a cell located at the destination has value 1, a cell located at the boundary of the search radius has value 0, and cells outside the search radius are clipped to 0. Thus, $D(\cdot)$ is dimensionless and bounded in $[0,1]$.

The parameter α is especially important from a user-centred perspective. Small values of α privilege proximity to the destination, while larger values privilege parking opportunity and thus potentially shorter search time. The simulation campaign in Section 4 shows that this is not a cosmetic parameter: it materially affects the trade-off between cruising time and final walking distance.

3.5. Heatmap Updates and Information Dynamics

The heatmap evolves through parking-related events. Each event is represented as

$$e = \langle \sigma_e, \ell_e, \tau_e, c_e \rangle, \tag{9}$$

where $\sigma_e \in \{\texttt{release}, \texttt{occupation}\}$ is the event type, ℓ_e is the event location, τ_e is the timestamp, and $c_e \in [0,1]$ is the event confidence. The confidence value can encode the reliability of the source event, for example the quality of the BLE state transition, the accuracy of the location estimate, or later trust-management mechanisms.

To incorporate spatial uncertainty, each event contributes not only to the cell in which it is observed, but also to nearby cells according to a spatial kernel. In the current formulation we use

$$K_\rho(\ell_e, \ell_{i,j}) = \exp\left(-\frac{d_{geo}(\ell_e, \ell_{i,j})^2}{2\rho^2}\right), \tag{10}$$

where $\ell_{i,j}$ is the centroid of cell (i,j), d_{geo} is the geographic distance between the event and the cell centroid, and ρ controls the spatial propagation radius.

Let $\mathcal{E}_{release}^{(t,t+\Delta t)}$ and $\mathcal{E}_{occupation}^{(t,t+\Delta t)}$ be the release and occupation events observed between t and $t+\Delta t$. The release and occupation evidence layers are updated as follows:

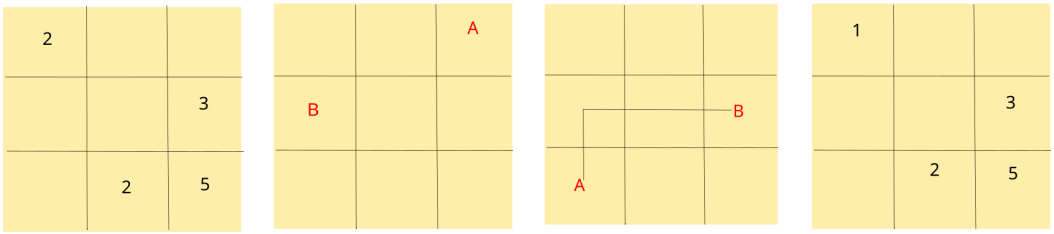
$$R_{i,j}^{(t+\Delta t)} = \exp(-\lambda \Delta t) R_{i,j}^{(t)} + \sum_{e \in \mathcal{E}_{release}^{(t,t+\Delta t)}} c_e K_\rho(\ell_e, \ell_{i,j}), \tag{11}$$

$$O_{i,j}^{(t+\Delta t)} = \exp(-\lambda \Delta t) O_{i,j}^{(t)} + \sum_{e \in \mathcal{E}_{occupation}^{(t,t+\Delta t)}} c_e K_\rho(\ell_e, \ell_{i,j}), \tag{12}$$

where $\lambda \geq 0$ is the temporal decay parameter. Recent events therefore have a stronger effect than older events, and spatially close events have a stronger effect than distant events.

The heatmap value used by the guidance policy is then obtained from Equations (4)–(6). This explicitly links the raw parking-related events, the heatmap evidence values, and the final occupancy/opportunity estimate used in the score function. A simplified ± 1 update can be recovered as a special case by setting $c_e = 1$, using a kernel that affects only the event cell, and setting $\lambda = 0$ over the considered update interval.

Figure 3 illustrates the qualitative idea of a heatmap evolving through repeated events. The figure is stylised, but it captures the core intuition: parking evidence accumulates spatially and gives agents a reason to favour some areas over others.



(**a**) Initial state (**b**) New event positions (**c**) Event propagation (**d**) Updated heatmap

Figure 3. Stylised evolution of the parking heatmap as new events are incorporated.

From a deployment viewpoint, this model is attractive precisely because it is lightweight. The system does not require an exact ground truth for every slot at every time. Instead, it relies on incremental evidence aggregation. This makes it compatible with the BLE-and-smartphone pipeline presented next.

3.6. From System Model to Prototype Implementation Counterparts

Table 4 summarises the correspondence between the main modelling assumptions and the prototype artefacts that support them. This table highlights the central accomplishment of the extension: the most important previously implicit assumptions now have a concrete implementation counterpart.

Table 4. How the main modelling assumptions map onto concrete prototype artefacts.

Model Assumption	Prototype Counterpart	Implication
Vehicles can generate parking-related events	BLE beacon connection/disconnection plus smartphone location	The event-generation assumption is implementable with commodity hardware
Parking evidence can be aggregated spatially	Server-side coordinate rounding and area-based querying	The heatmap abstraction has a practical server-side realisation
Drivers can inspect or exploit a shared parking representation	Map-based mobile interface	Shared information can be made available through an ordinary smartphone application
Information quality depends on participation and prior evidence	Event database populated over time	Simulation parameters such as adoption and initialisation correspond to real operational conditions
Guidance can prioritise opportunity versus distance	Score-based policy from the simulation, ready for integration	The missing step is integration of the decision engine, not reinvention of the whole system

The table makes clear that the remaining gap is relatively narrow and well defined. The prototype already supports event capture, storage, aggregation, and map visualisation. The simulation already supports the decision logic. The next stage of the research is therefore to integrate the two.

4. Simulation-Based Evaluation

We now present the simulation campaign used to evaluate the cooperative parking-guidance strategy. The experiments analyse how shared heatmap information affects parking search under different levels of participation, initial information availability, and user-preference settings. This broader perspective allows us to ask a more deployment-relevant question: If a system like the prototype can collect parking-related evidence, how much participation is needed before that evidence becomes useful enough to improve parking search?

The simulation framework and the prototype implementation share the same event-driven state model. The simulation abstracts the event detection process, while the prototype replaces the abstract event generator with the BLE-based state detection module. In the simulation, state transitions are generated according to predefined vehicle parking and departure events, while in the prototype the same transitions are triggered by the BLE connection and disconnection states. Therefore, the simulation and the prototype operate on the same event semantics, differing only in the mechanism used to generate the events.

4.1. Simulation Environment and Experimental Setup

The experiments were conducted in SUMO, controlled in real time via TraCI [27,28]. Each simulated vehicle acted as an autonomous agent. Some agents exploited the shared heatmap when searching for parking, while others behaved as non-cooperative baseline vehicles. The synthetic road network used in the experiments reproduced the topology of a small urban environment with intersections, neighbourhood-scale connectivity, and distributed parking areas (Figure 4).

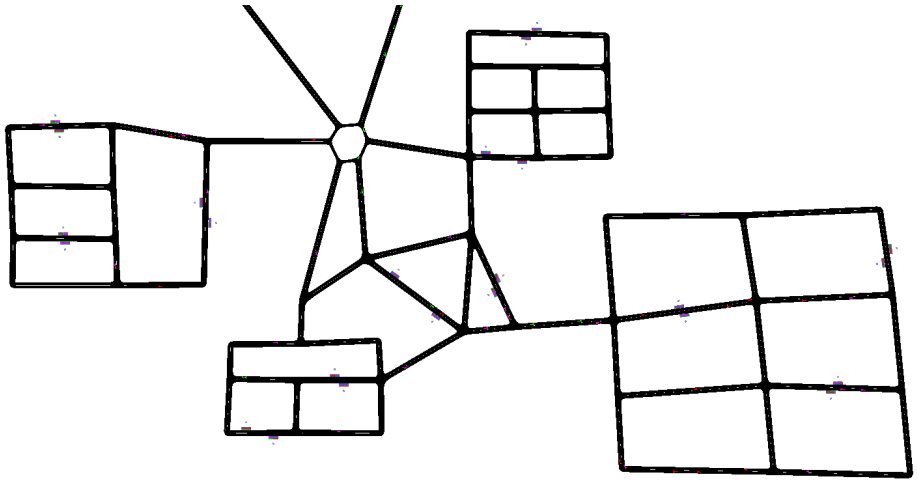


Figure 4. Synthetic SUMO road network used for the agent-based simulation campaign. Although synthetic, the network captures the topology of a small urban area with several competing parking zones.

The most relevant experimental parameters are summarised in Table 5. Each configuration was executed ten times and the results were averaged. The total number of vehicles was fixed at 300, while three families of variables were explored: the adoption rate of the heatmap, the value of α, and the amount of initial heatmap information.

A final methodological clarification is useful here. The simulation is not meant to predict the exact behaviour of a real city. Rather, it serves as a controlled environment for understanding the behavioural consequences of information sharing. This is exactly why the prototype introduced in Section 5 is so valuable: it complements the simulation by grounding the data-collection assumptions in an implementable system.

Table 5. Main simulation parameters and performance indicators.

Parameter	Values/Description
Traffic simulator	SUMO with TraCI-based interaction
Agents	300 simulated vehicles
Repetitions per configuration	10 runs
Heatmap adoption rate	From 0% to 100% of vehicles using the cooperative guidance strategy
Preference parameter α	Values between 0.2 and 0.8
Initial heatmap information	From 0% to 100% initialisation
Baseline	Agents without heatmap support that begin searching only after reaching the destination area
Primary metrics	Average parking search time; distance between parking position and destination
Evaluation focus	Practical usefulness under partial participation and different information availability levels

4.2. Model Calibration and Validation

The calibration and validation considered in this study concern the scientific behaviour of the cooperative guidance model rather than the functional validation of the prototype. The model contains both *behavioural parameters* and *deployment-dependent parameters*. In particular, α represents a user preference rather than a parameter fitted to maximise system performance; accordingly, it is varied over the interval considered in the experiments rather than assigned a single optimal value. Similarly, the adoption rate and the amount of initial heatmap information are treated as experimental factors. This parameter sweep provides a sensitivity analysis showing how the model responds when its principal behavioural and information-related assumptions are changed.

The mathematical formulation also provides several internal consistency checks. Both $H_{score}(i, j, t)$ and $D(\cdot)$ are dimensionless and bounded in $[0, 1]$, so Equation (3) is a convex combination, and $S(i, j)_T^{(t)}$ is likewise bounded in $[0, 1]$. The limiting cases have a direct interpretation: $\alpha = 0$ produces a proximity-only decision, whereas $\alpha = 1$ produces an opportunity-only decision. Moreover, increasing release evidence increases the estimated parking opportunity, increasing occupation evidence reduces it, and the confidence factor limits the influence of cells supported by little evidence. The spatial kernel assigns greater influence to geographically closer observations, while the exponential decay ensures that older observations progressively lose weight. These properties provide analytical checks that the model behaves consistently with its intended semantics.

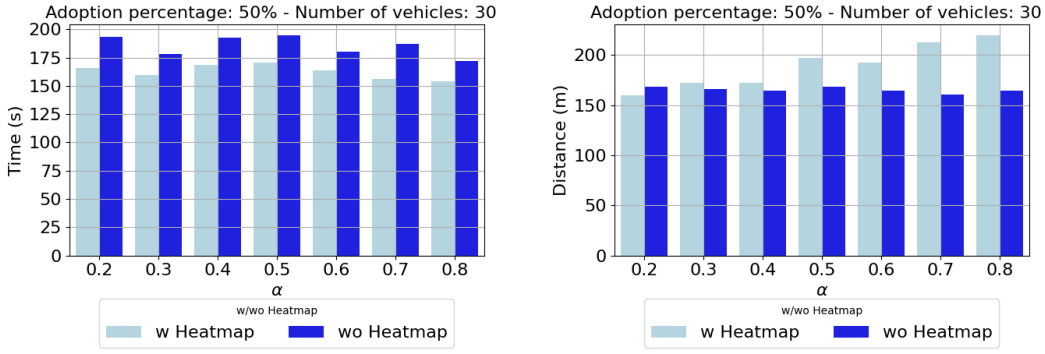
The simulation provides a complementary behavioural validation. The model is compared with a non-cooperative baseline and evaluated through repeated runs while independently varying user preference, system adoption, and initial information availability. The resulting trends are consistent with the mechanisms encoded by the model: changing α shifts the trade-off between search time and walking distance, increased participation modifies the usefulness and freshness of shared information, and greater initial information mitigates the cold-start condition. These experiments therefore support the internal and behavioural validity of the proposed guidance mechanism under controlled conditions.

The parameters associated with the generalised evidence-update formulation, including the spatial propagation parameter ρ, temporal decay parameter λ, event confidence c_e, and smoothing constant ε, were not calibrated against a real-world parking dataset

in the present study. Their appropriate values are expected to depend on deployment-specific properties such as positioning uncertainty, parking turnover, sensing reliability, and observation density. Consequently, the present evaluation establishes the mathematical consistency and controlled behavioural validity of the model but does not claim empirical calibration or predictive validity for a specific city. Such calibration requires field observations against which these parameters and the resulting parking-opportunity estimates can be compared.

4.3. Effect of the Opportunity–Distance Trade-Off

We first analysed the effect of the preference parameter α. Figure 5 reports the average parking search time and average parking distance for a representative configuration with 50% heatmap adoption and 20% initial heatmap information. The pattern is intuitively meaningful. When α increases, agents place more weight on the opportunity component and less on destination proximity. As a consequence, the average parking distance tends to increase, because agents are more willing to accept a farther spot if it appears promising. At the same time, average search time tends to decrease or remain competitive, because agents avoid overly stubborn local cruising near the destination.



(**a**) Average parking search time (**b**) Average parking distance

Figure 5. Effect of the parameter α with 50% heatmap adoption and 20% initial heatmap information.

This result is interesting for at least two reasons. From a modelling viewpoint, it confirms that the cooperative strategy is not tied to a single user preference profile. From a deployment viewpoint, it suggests that a future integrated system could expose different recommendation modes, for example "park closer" versus "park faster", without altering the underlying information infrastructure.

A second comparison, shown in Figure 6, highlights the same trade-off in a configuration with 75% participation and 60% initialisation by directly contrasting the heatmap-guided strategy with the no-heatmap baseline. Even when the gains are not monotonic for every value of α, the cooperative strategy remains competitive and typically superior in average search time.

4.4. Effect of Participation Under Cold and Warm Heatmaps

The central deployment question concerns adoption. A real parking-support system is unlikely to reach universal participation immediately, especially in its early phases. We therefore examined how performance varied as the fraction of cooperative vehicles grew.

Figure 7 reports the effect of adoption when the heatmap starts from a cold state, that is, with no initial information. In this case, each additional participating vehicle contributes directly to the collective knowledge base. The figure shows that average parking search time decreases as adoption increases, which is exactly what one would hope from a cooperative

system. Parking distance exhibits a more nuanced behaviour, because high participation also means that more agents may be attracted to the same apparently promising areas.

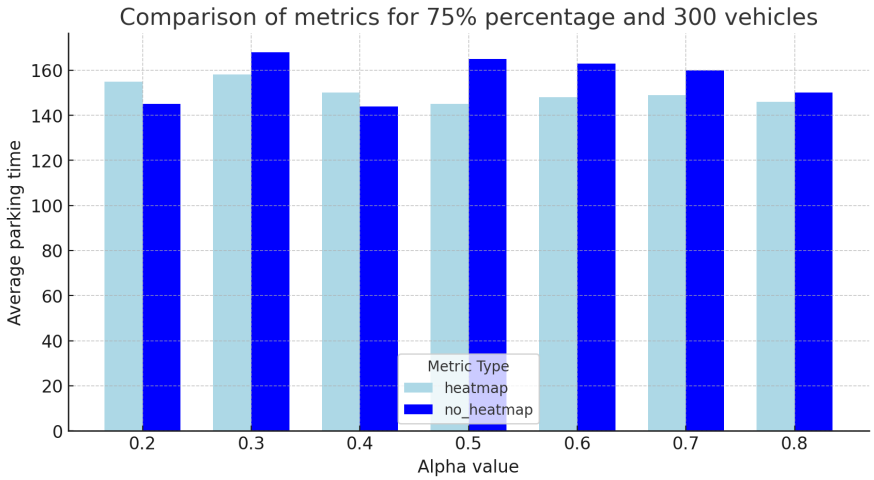
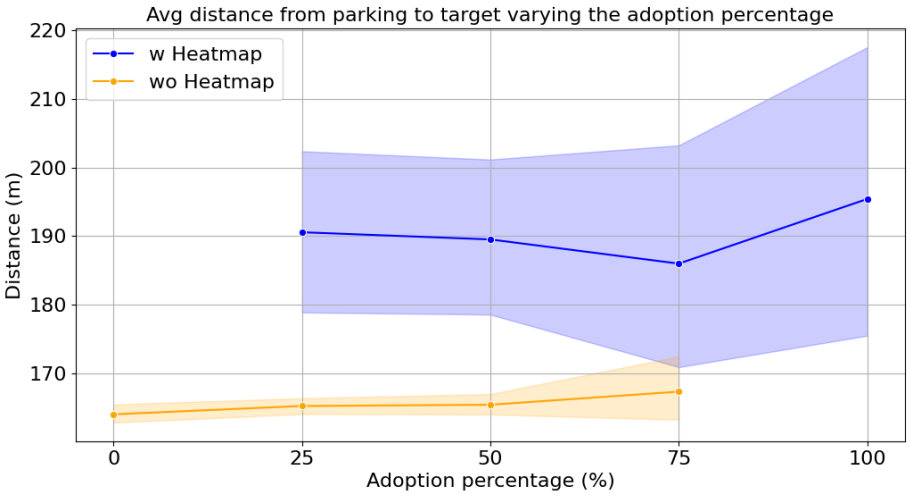
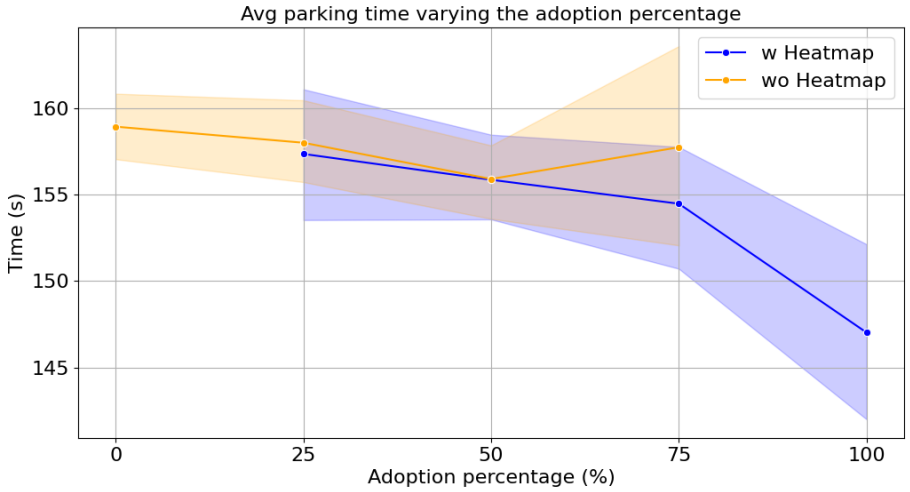


Figure 6. Comparison between heatmap-guided and non-cooperative search for different values of α in a configuration with 75% participation and 60% initialisation.



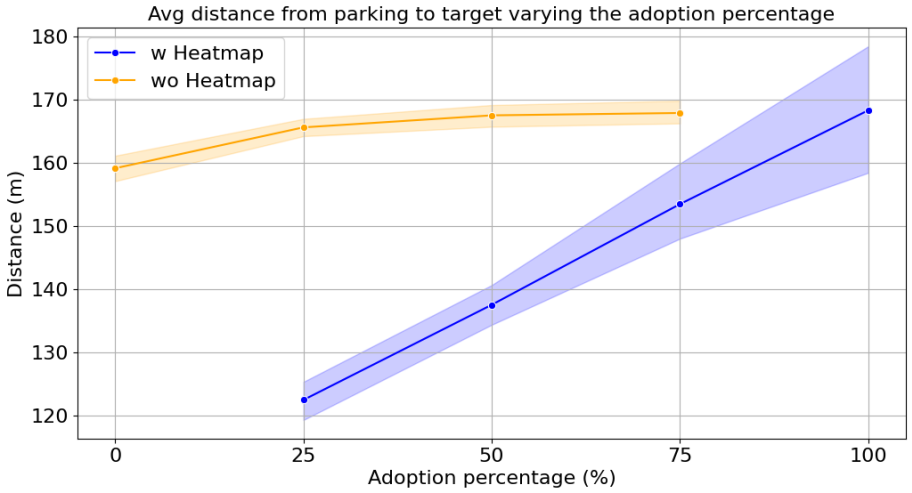
(**a**) Average parking distance



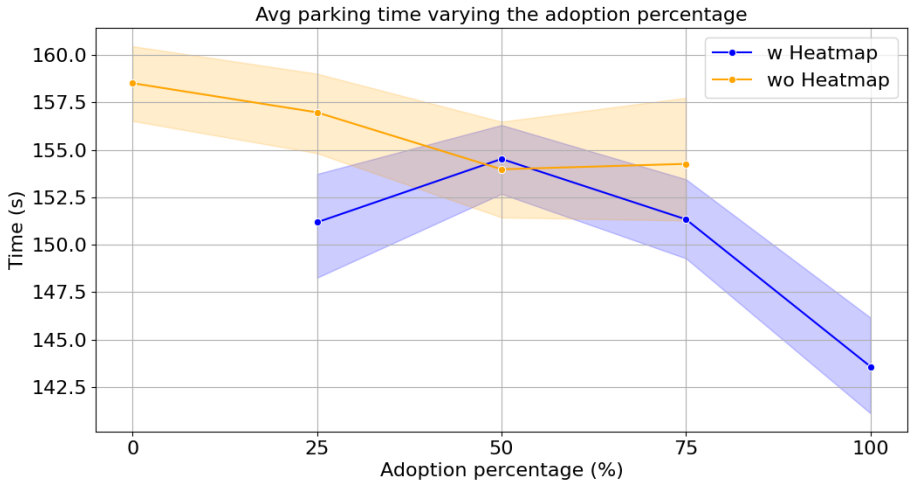
(**b**) Average parking search time

Figure 7. Effect of adoption when the heatmap is cold (0% initial information). Shaded areas are the standard deviations.

Figure 8 considers the opposite extreme, in which the heatmap starts from a warm state with abundant initial information. Search time again benefits from increased participation, which indicates that cooperative guidance remains useful even when the system is not information-starved at the start. However, the spatial competition effect becomes more visible: when many agents receive similar guidance, some of them converge toward the same attractive regions.



(**a**) Average parking distance



(**b**) Average parking search time

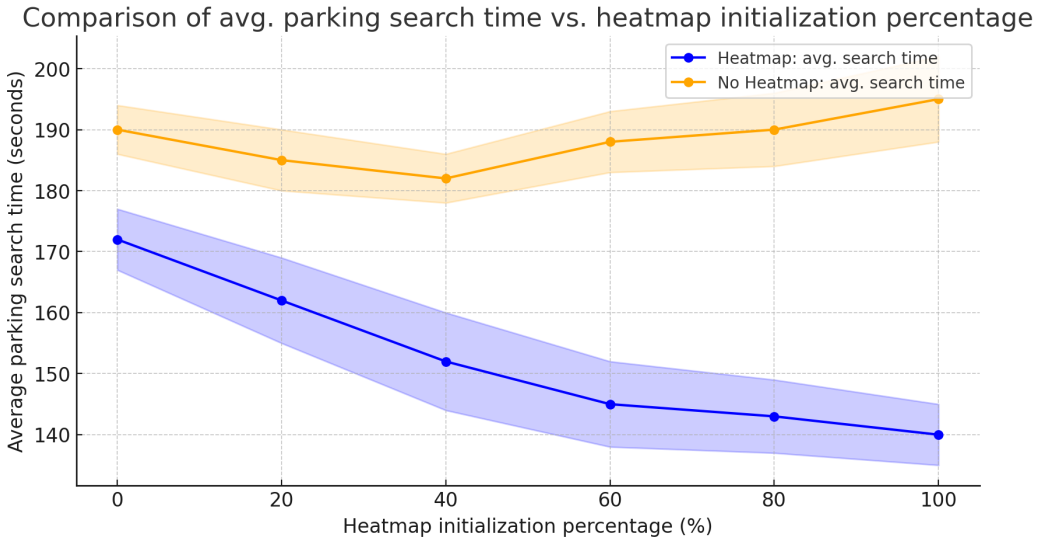
Figure 8. Effect of adoption when the heatmap is warm (100% initial information). Shaded areas are the standard deviations.

Taken together, the cold and warm cases tell a useful story for deployment. In early adoption phases, each new user is valuable because they generate information. In later phases, once the heatmap becomes informative, recommendation quality becomes increasingly coupled with the system's ability to manage correlated demand.

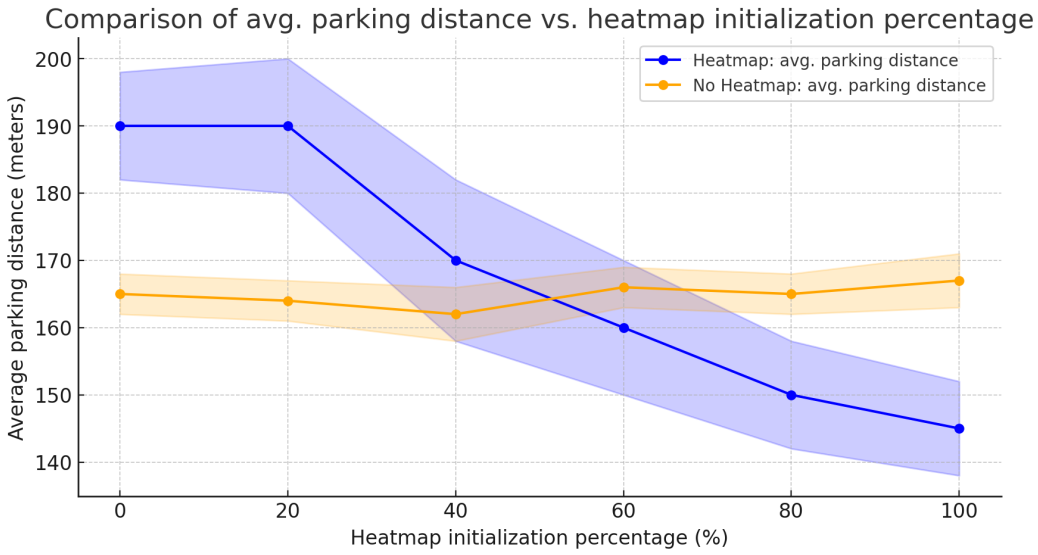
4.5. Effect of Initial Heatmap Information

The next question is how much pre-existing information matters. Figure 9 isolates the effect of heatmap initialisation. The pattern is clear: more initial information generally leads to better outcomes, especially in terms of parking search time. This is highly relevant for an operational system, because it means that historical or already accumulated daily evidence can materially improve the usefulness of the guidance process.

This observation is one of the clearest bridges between the simulation and the prototype. In the simulation, initialisation is a parameter. In the prototype, initialisation can be interpreted concretely as the amount of already-stored evidence available when the user begins a search. In other words, what appears in the simulation as a modelling condition maps in the prototype to the practical value of historical and recently contributed data.



(**a**) Average parking search time



(**b**) Average parking distance

Figure 9. Effect of the initial amount of heatmap information on cooperative parking performance. Shaded areas are the standard deviations.

4.6. Integrated Reading of the Quantitative Results

Figure 10 offers an additional integrated perspective by focusing on a high-participation configuration. Although the detailed values are less important than the overall trend, the figure reinforces the general message already emerging from the previous plots: parameter tuning meaningfully shifts the balance between distance and time, and the cooperative strategy remains sensitive to the trade-offs encoded in the decision function.

Table 6 summarises the main interpretive findings of the simulation campaign.

Table 6. Synthesis of the most relevant findings from the simulation campaign.

Question	Main Finding and Relevance
Effect of α	Increasing α generally prioritises parking opportunity over proximity, often reducing search time at the cost of larger walking distance; this supports preference-aware future recommendations
Effect of adoption	Search time tends to improve as more vehicles participate, indicating that meaningful benefits emerge even before full adoption
Effect of warm vs. cold start	Cold heatmaps benefit strongly from additional users because each participant also contributes information; warm heatmaps remain beneficial but show stronger competition effects
Effect of initialisation	Pre-existing information materially improves performance, especially for search time; this directly supports using historical or same-day evidence in a deployment
Overall implication	Cooperative parking guidance is robust enough to be interesting under partial adoption and sufficiently grounded to justify integration with the prototype pipeline

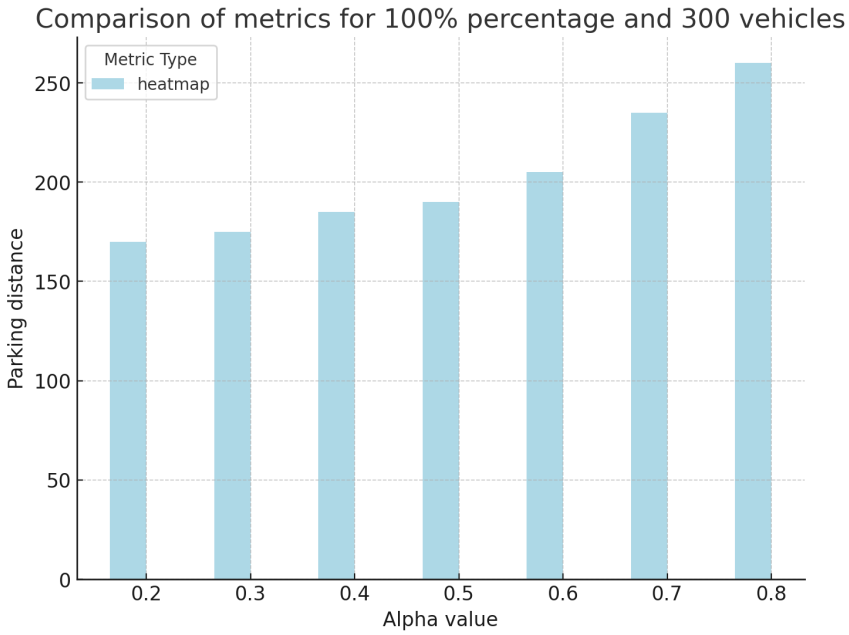


Figure 10. Illustrative comparison of parking-distance outcomes for different values of α in a full-participation configuration.

5. Proof-of-Concept Realisation of the Parking-Evidence Pipeline

The simulation study evaluated the behavioural performance of the cooperative guidance model, but it assumed that parking-related evidence could be generated and made available to participating vehicles without dedicated roadside sensing infrastructure. This section examines the implementation feasibility of that assumption through a compact proof-of-concept realisation of the corresponding data pipeline. The prototype demonstrates how BLE-based user–vehicle proximity transitions can be converted into geolocated parking-related events, transmitted to a back-end service, stored, spatially aggregated, and visualised as shared evidence. Its purpose is limited to establishing the feasibility of this end-to-end information-generation process; it is not intended to validate the parking-guidance strategy or to establish real-world sensing accuracy, reliability, latency, or deployment readiness.

5.1. Mapping Model Assumptions to the Prototype

The prototype implements the minimum set of functions required to examine whether the parking-evidence mechanism assumed by the model can be realised without fixed roadside sensing infrastructure. Four functional elements are involved: an in-vehicle BLE beacon provides a proximity signal; the smartphone detects relevant state transitions and associates them with the current location; a back-end service receives and stores the resulting events; and stored events are spatially aggregated for visualisation as shared parking evidence.

Table 7 relates these implemented functions directly to the assumptions introduced in the cooperative parking model. The purpose of the mapping is not to prescribe a specific software stack, but to show how the information required by the model can be obtained through commodity components.

Table 7. Correspondence between model assumptions and their proof-of-concept realisation.

Model Requirement	Prototype Realisation	Function Demonstrated
Detection of parking-relevant state changes	BLE beacon and smartphone	Uses changes in user–vehicle proximity as triggers for generating parking-related evidence.
Association of evidence with location	Smartphone positioning	Associates each detected state transition with the location at which the event is generated.
Communication and persistence of evidence	Mobile-to-server communication and back-end storage	Transfers timestamped parking-related events to a shared service for subsequent retrieval and aggregation.
Construction of a shared spatial representation	Server-side spatial aggregation	Groups stored events geographically to produce the evidence points required for heatmap construction.
Availability of shared evidence to participating users	Map-based visualisation	Demonstrates that aggregated parking evidence can be retrieved and presented for a selected geographic area.

Only state transitions are reported rather than continuously transmitting an unchanged state. This avoids redundant observations and preserves the event-based semantics used by the model. The implementation stores each event together with its location and timestamp, allowing the collected evidence to support spatial aggregation and, in principle, the temporal weighting mechanisms defined in Section 3.

The proof of concept assumes sufficiently stable BLE communication between the in-vehicle beacon and the smartphone to detect user–vehicle proximity transitions. Because transient interference, operating-system power management, or signal fluctuations may produce short-lived connection changes, the application applies a simple temporal filter and accepts a state transition only when the new state persists for a predefined interval. This mechanism is intended to reduce spurious events; however, BLE detection accuracy and robustness were not systematically evaluated in the present study and therefore remain subjects for future field validation.

5.2. BLE Pairing and User Interaction Workflow

The first requirement of the prototype is to pair the app with the in-vehicle beacon. The workflow differs slightly depending on whether the app is being launched for the first time or whether a beacon has already been associated with the user. Figure 11 illustrates the first-run scenario, while Figure 12 shows the behaviour on later launches.



Figure 11. First-run interaction flow: the user scans nearby BLE devices, selects the beacon installed in the vehicle, and saves it locally for subsequent use.

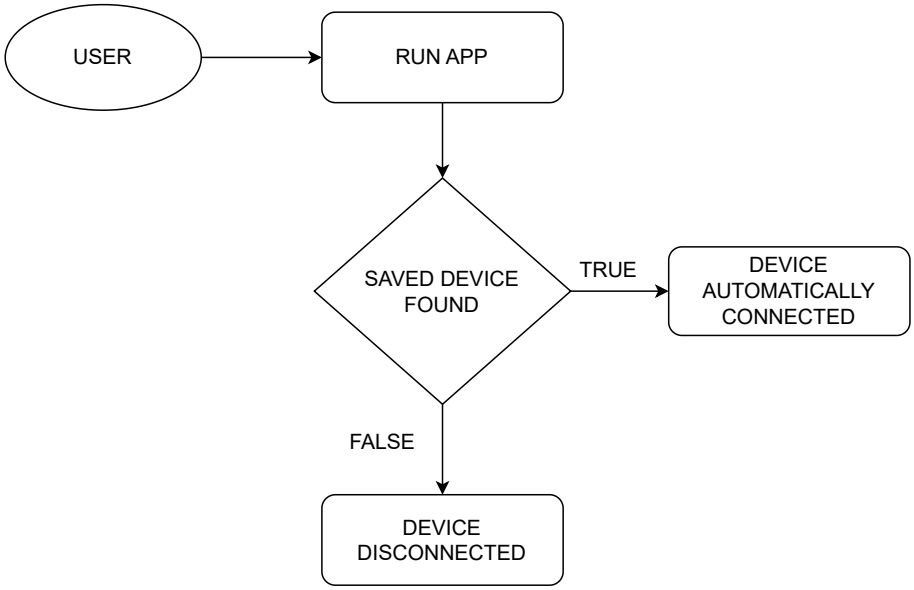


Figure 12. Subsequent interaction flow: if the saved beacon is found, the application reconnects automatically; otherwise the user remains disconnected until the beacon becomes available again.

This pairing logic is more than a user-interface detail. It ensures that the system remains usable in everyday conditions. The first interaction requires deliberate user action, but subsequent operation becomes mostly automatic. That is important because infrastructure-less systems only scale if the reporting burden placed on users is minimal.

At a lower level, the app performs periodic BLE scans, filters devices, and stores the selected device identifier in local persistence. On subsequent launches, the app can attempt reconnection immediately. When the connection status changes, the application interprets the transition as a parking-relevant event.

5.3. From State Changes to Server-Side Events

Once the smartphone has been associated with the vehicle beacon, changes in BLE proximity are interpreted as parking-relevant state transitions. Rather than continuously transmitting the current state, the application reports only transitions, thereby preserving the event-based semantics introduced in the cooperative parking model.

Figure 13 summarises this process. When a persistent state change is detected, the application obtains the current position and creates a timestamped event containing the corresponding state and geographic coordinates. The event is then transmitted to the back-end service and stored for subsequent spatial aggregation. In this way, the prototype provides a concrete implementation of the model's assumption that participating users can contribute geolocated parking-related evidence without dedicated roadside sensing infrastructure.

5.4. Area-Based Requests and Aggregation Pipeline

The second part of the prototype concerns retrieval and spatial aggregation of the collected evidence. When the user changes the geographic area displayed on the map, the application issues a request for parking-related events within the corresponding spatial and temporal region. Figure 14 summarises the communication between the mobile application, back-end service, and event store.

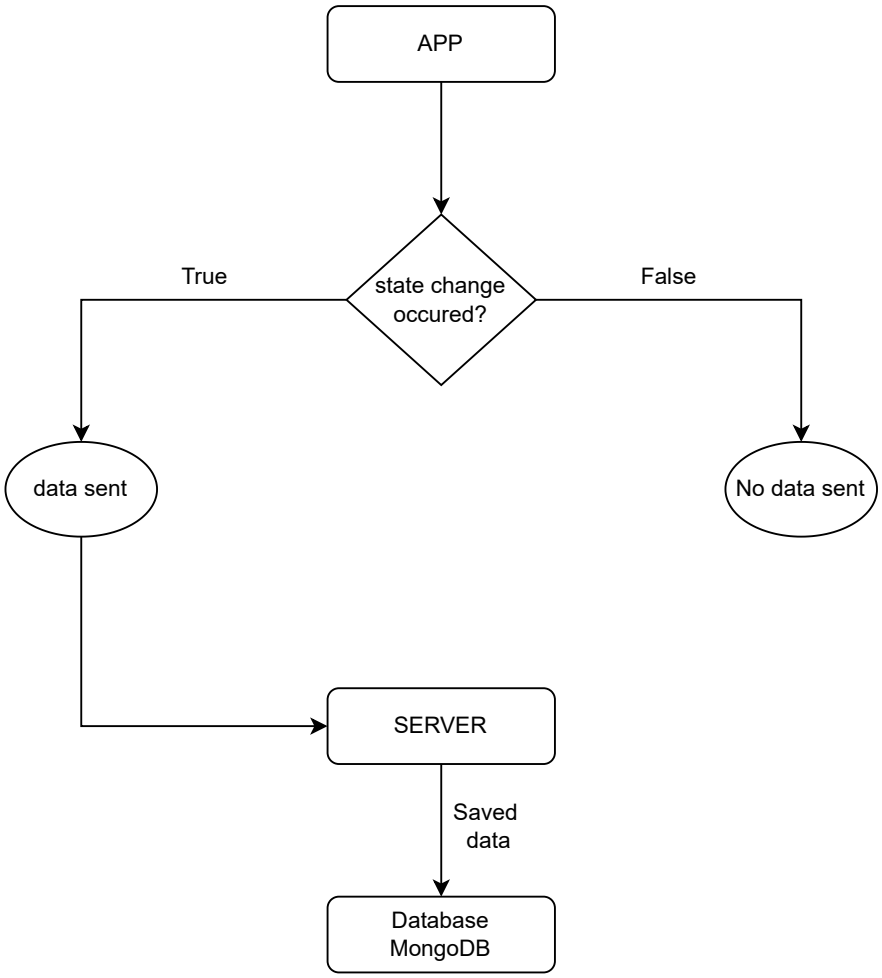


Figure 13. Application-side logic for sending data to the server. Only state changes are transmitted. The initial state is used locally to initialise behaviour but is not itself recorded as an event.

The request specifies a temporal interval and a geographic area. The back end retrieves the corresponding events and aggregates them according to location and state. In the current proof of concept, nearby observations are grouped using rounded coordinates, producing heatmap points that contain an approximate position, event count, and event type. Coordinate rounding is used only as a lightweight baseline for demonstrating the aggregation and visualisation pipeline; it is not an intrinsic requirement of the proposed cooperative guidance method.

This process provides an implementation counterpart to the spatial evidence representation used by the model. Although the prototype currently visualises state-aware event counts, each stored observation retains its state, location, and timestamp. The resulting event structure can therefore support more advanced aggregation mechanisms, including the confidence weighting and temporal decay defined in Section 3. Alternative spatial aggregation methods, such as adaptive grids, density-based clustering, or kernel-based approaches, can likewise replace coordinate rounding without modifying the underlying event-acquisition pipeline.

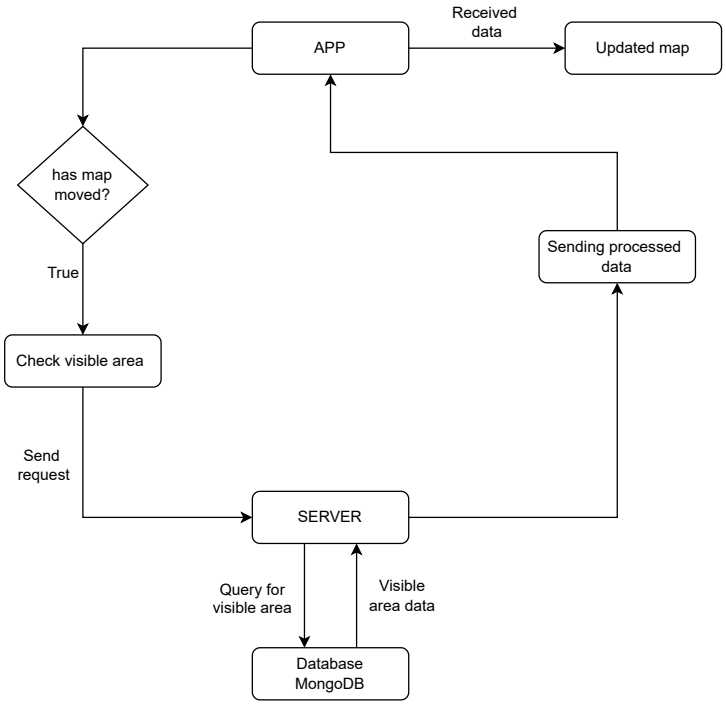


Figure 14. Data flow between the mobile application, the back-end server, and the database when parking evidence is requested for a geographic area.

5.5. Front-End and Back-End Implementation Choices

The prototype was implemented as a cross-platform mobile application using React Native (v 0.74) and Expo (v 51.0), combined with a Node.js/Express (v 4.21) back end and MongoDB persistence. These technologies were selected for pragmatic reasons.

First, a cross-platform mobile framework allows the same codebase to target both Android and iOS, which is desirable if the system is to be used by an ordinary driver population rather than by a narrowly constrained testbed. Second, the app framework integrates well with geolocation, HTTP communication, local storage, and BLE management, all of which are essential to the application logic. Third, the back-end stack keeps the server lightweight and easy to deploy, which is consistent with the low-cost philosophy of the project.

From a software-architecture standpoint, the prototype is organised around four modules:

- A **BLE management module**, responsible for scanning nearby devices, selecting the in-vehicle beacon, handling reconnection, and exposing connection status changes;
- A **location and map module**, responsible for acquiring the user position, controlling map movement, resolving search queries, and determining the visible region;
- A **networking module**, responsible for posting state-change events and requesting aggregated heatmap data for a region of interest (see Appendix A for event messages details);
- A **server-side aggregation module**, responsible for validating requests, filtering the event store by time and space, and grouping similar coordinates into map points.

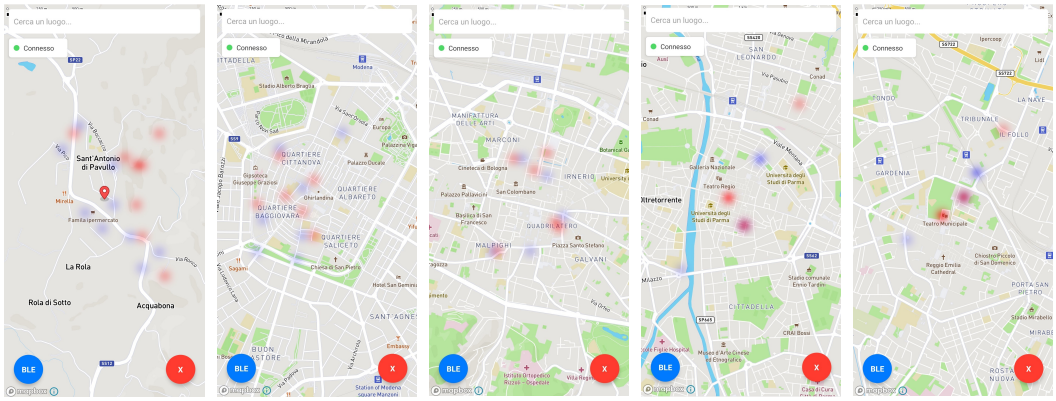
This modularisation matters because it maps cleanly onto future extensions. For example, replacing the current coordinate-rounding strategy with a clustering or probabilistic occupancy layer would affect the aggregation module but not the BLE or map components. Likewise, introducing user authentication or privacy-preserving identifiers would affect the networking and persistence layers without changing the application-level state semantics.

5.6. Prototype Functional Testing and Feasibility Assessment

The prototype was evaluated through functional tests rather than through a field trial. These tests were designed to verify that the implemented components executed the intended end-to-end workflow: BLE-related state transitions could be processed, events could be transmitted and stored, geographic queries could be performed, and aggregated evidence could be rendered on the map. The tests did not quantify parking-state detection accuracy, communication latency, reliability, energy consumption, robustness under prolonged operation, or user-level performance. Consequently, this assessment supports a technical-feasibility claim rather than a claim of validated real-world performance or deployment readiness.

Two sets of tests were particularly useful. The first populated the back end with simulated events in different cities to verify that the app correctly queried and rendered heatmap layers over distinct geographic contexts. The second manipulated the density of events at fixed coordinates to check whether the heatmap intensity changed as expected and whether overlapping evidence of different states remained visually interpretable.

Figure 15 shows representative screenshots from these tests. Within the tested configurations, the results show that the map interface, geographic filtering, and state-aware heatmap-rendering functions operated according to the intended workflow across the selected geographic contexts.

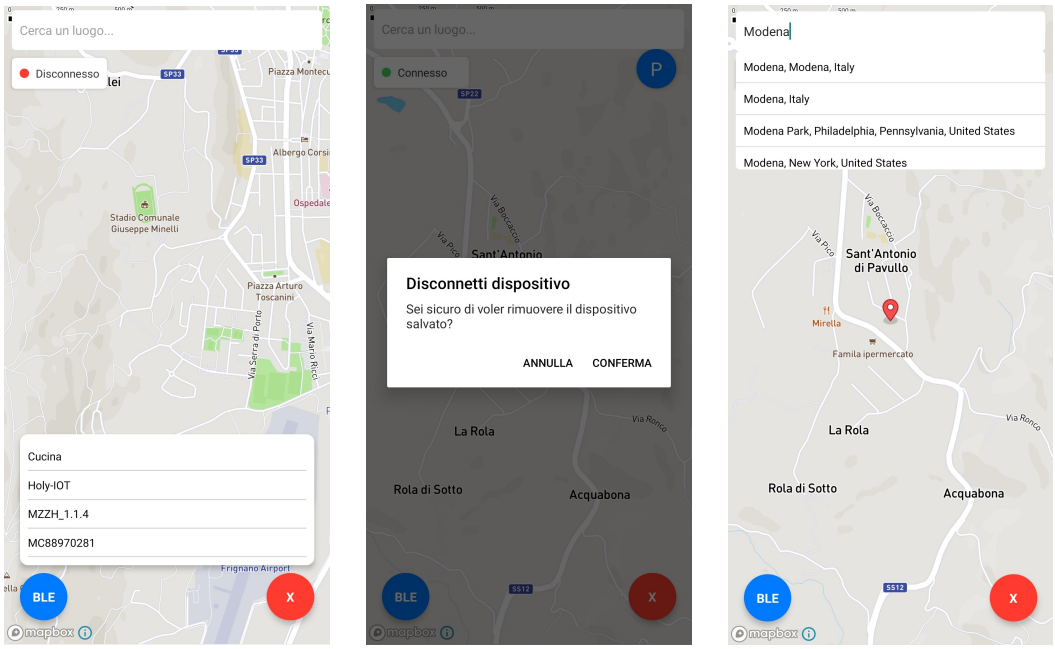


(**a**) User location (**b**) Modena (**c**) Bologna (**d**) Parma (**e**) Reggio Emilia

Figure 15. Representative screenshots from the prototype application. Different test cities and regions confirm that the server query, spatial filtering, and state-aware heatmap rendering pipeline behaves coherently. "Connesso" means Connected; "Cerca un luogo" means "Search a place".

Figure 16 complements these screenshots with a closer look at specific interface elements used during the functional tests, including device discovery, device removal, and the map search interface.

Taken together, these tests support a limited but important feasibility claim: the data-acquisition and visualisation assumptions underlying the simulation can be instantiated in a proof-of-concept software pipeline using inexpensive BLE hardware, an ordinary smartphone, and a lightweight server-side stack. The tests demonstrate end-to-end functional integration, but they do not establish the accuracy of BLE-derived parking-state evidence or the performance of the system under real-world operating conditions.



(**a**) BLE device list (**b**) Removing a saved device (**c**) Search interface

Figure 16. Additional user-interface elements of the prototype used during the validation process. "Disconnesso" means Disconnected; "Cerca un luogo" means "Search a place"; The popup in second image says "Do you want to remove the device? Cancel and Confirm"

5.7. Role of the Prototype in the Proposed Approach

The prototype does not yet implement the full score-based recommendation policy from Equation (3). At present, it focuses on data acquisition, storage, filtering, aggregation, and visualisation. This is an important and deliberate scope choice. Before a full recommendation engine can be deployed, the system must first demonstrate that parking-related evidence can be generated automatically and transformed into a useful spatial representation.

The prototype therefore demonstrates the technical feasibility of implementing the principal stages of the data pipeline required by the cooperative guidance model. It shows that BLE connectivity transitions can be captured and treated as parking-related evidence, combined with geolocation, transmitted to a back end, stored as timestamped events, aggregated by area, and visualised as heatmap evidence. This provides an implementation baseline for subsequent integration of the score-based guidance policy. It does not, however, validate BLE connectivity as an accurate indicator of parking behaviour against real-world ground truth.

6. Operational Workflow and Engineering Rationale

The previous section introduced the prototype mainly from an architectural perspective. In this section, we look at the same system from an operational viewpoint. This is useful for two reasons. First, it clarifies how the application behaves during ordinary usage, which is important for understanding the realism of the infrastructure-less assumption. Second, it explains why several apparently small engineering decisions are actually central to the quality of the resulting parking evidence.

6.1. Vehicle Presence as a Low-Cost Sensing Proxy

The key design decision behind the prototype is to use vehicle presence rather than direct slot sensing. In conventional smart-parking deployments, a sensor is usually associated with the parking resource itself. In our case, the low-cost beacon is associated with the

vehicle, and the smartphone is associated with the user. The parking-related interpretation emerges from the relationship between the two.

Figure 17 illustrates the intuition. When the smartphone remains in proximity to the beacon, the system interprets the user as being in the vehicle. When the smartphone moves away and the BLE connection is lost, the system interprets this as evidence that the vehicle has been parked. Conversely, when a previously disconnected smartphone reconnects to the beacon, the system interprets this as evidence that the driver has returned to the vehicle and may free a parking spot shortly afterwards.

This design has several advantages. It avoids installing hardware in each parking position. It works in open parking areas where ownership and physical structure are heterogeneous. It also shifts the sensing cost from the city or infrastructure operator to a simple, inexpensive device that can travel with the vehicle. In practice, this means that scaling the system depends more on user adoption than on civil-engineering deployment.

The design also has limitations. BLE proximity is not equivalent to certain knowledge of parking status. A user may leave the smartphone in the vehicle, may briefly move in and out of range, or may interact with the application in unexpected ways. Nevertheless, as a low-cost signal, it is strong enough to justify the event model used by the prototype and the simulation. The paper does not claim that the signal is perfect; it claims that it is good enough to support a scalable evidence-collection mechanism.



(**a**) Connected state: user near the vehicle (**b**) Disconnected state: user moved away from the vehicle

Figure 17. BLE proximity as a low-cost proxy for parking-related state changes. The beacon does not sense the parking slot directly; instead, the connection state between beacon and smartphone is used to infer whether the vehicle is in motion or parked.

6.2. Application Lifecycle and Event Discipline

A second important design decision concerns *when* the app should transmit information. It is tempting to think that sending more data is always better, but in this case, indiscriminate transmission would reduce data quality instead of increasing it. The application therefore follows a disciplined lifecycle.

On first use, the user explicitly scans for nearby BLE devices and selects the beacon to be associated with the vehicle. Once selected, the beacon identifier is stored locally. On later launches, the app attempts to reconnect automatically. This behaviour is significant because it turns the long-term user experience into a mostly passive one. The system would be much less plausible as an everyday tool if the user had to repeat the pairing procedure at every launch.

The initial state at application startup is used only to initialise behaviour. It is not transmitted to the back end. The reason is methodological as much as technical. At startup, the application may be in a transient situation: permissions are still being checked, the device may not yet have been discovered, and location may still be stabilising. Treating this transient situation as a parking event would pollute the database with ambiguous reports. By transmitting only subsequent changes, the app records events that are more tightly coupled to an actual parking-relevant action.

This event discipline also improves the interpretability of the server-side history. A sequence of state transitions corresponds to a sequence of meaningful changes in the relationship between user and vehicle. If the system were to send repeated steady-state reports,

the event log would be harder to interpret and the heatmap could become dominated by redundant evidence. In other words, the application behaves like an *event source*, not like a continuous telemetry sensor.

A related decision concerns map-triggered requests. The app does not request updated data continuously during every small map movement. Instead, it waits until the user has meaningfully moved, zoomed, or searched for a new area. This reduces network traffic and prevents the back end from being flooded with near-identical requests. Again, the choice is modest at the engineering level but central at the system level: a deployable architecture must control both the quality and the volume of its data flows.

6.3. Spatial and Temporal Aggregation Logic

The back end receives two different kinds of inputs: parking-related events generated by the application, and user-driven requests for heatmap data. The first are stored; the second are answered by querying and aggregating the stored event history.

The temporal filtering strategy in the current prototype is deliberately simple. The application requests data within a day-bounded time window. This decision keeps the visualisation manageable and reduces the risk that the interface becomes dominated by stale evidence accumulated over long periods. While the current daily window is a design choice rather than a theoretical necessity, it reflects an important principle: parking information should be recent enough to remain operationally useful.

The spatial filtering strategy is equally pragmatic. The user specifies an area only indirectly, through their interaction with the map. The application identifies the centre of the visible region and transmits it together with a radius. On the server, that circular area is converted into a rectangular bounding box because bounding-box queries are more convenient for the database layer. After retrieval, the current prototype rounds event coordinates according to a chosen granularity so that nearby points are grouped together. This is a simple baseline aggregation strategy used for functional validation and visualisation, not a methodological constraint of the overall approach. The server also keeps parked and departing evidence distinct, because the prototype is designed to show these two kinds of evidence as separate though potentially overlapping heatmap layers.

This aggregation pipeline has two desirable properties. First, it is computationally cheap. Querying a rectangular region and grouping rounded coordinates are simple operations that can be executed quickly even on a lightweight back end. Second, it preserves a direct conceptual relationship with the simulation model. The simulation does not reason over raw GPS points either; it reasons over an aggregated spatial representation.

At the same time, coordinate rounding is only a first implementation choice. In dense urban environments, fixed rounding may create artificial boundaries or merge spatially distinct parking situations. A more advanced deployment can replace this step with density-aware clustering, adaptive grid cells, or kernel-based spatial aggregation. This replacement would affect only the aggregation module, while leaving the BLE event-collection pipeline, the database schema, and the map-based interaction model unchanged.

6.4. An End-to-End Usage Scenario

To make the overall flow more concrete, it is useful to describe a complete usage scenario. Consider a driver who has already paired the smartphone application with the beacon installed in the vehicle. The driver departs from one part of the city and reaches a destination in another area. While the driver is inside the vehicle, the smartphone remains connected to the beacon. When the driver parks and walks away, the connection is lost. The app detects the transition, acquires the current geographic information, and transmits a `parked` event to the server. The server stores the event with a timestamp.

Later, a second user opens the application and searches for parking near the same district. The user pans the map toward that area, which triggers a bounded request. The server retrieves all matching events in the relevant time window, aggregates them by coordinates and state, and returns the corresponding heatmap points. The application displays these points on the map. The second driver does not yet receive a route or a ranked recommendation in the current prototype, but they do receive a useful, evidence-based visual cue about where parking-related activity has recently been observed.

This scenario highlights the practical role of the proposed architecture. The heatmap is not treated as an abstract simulation artefact only; it is produced through an explicit and implementable software process based on user devices, server-side storage, spatial aggregation, and map-based visualisation.

6.5. Design Trade-Offs and Future Engineering Choices

The prototype also makes several future engineering choices more visible than before. One concerns stale data. A daily time window is a reasonable first compromise, but a real deployment will likely need recency weighting, confidence intervals, or event decay. Another concerns spatial granularity. If the aggregation cells are too coarse, the heatmap becomes vague; if they are too fine, the map becomes noisy and sparse. A third concerns energy consumption and mobile operating-system policy. BLE scanning, location access, and background execution are all affected by platform-specific constraints, so practical deployment will require careful optimisation.

A further issue concerns privacy and trust. The system is attractive partly because it does not need roadside infrastructure, but that shifts responsibility toward user devices and server-side data stewardship. A production-quality version would need stronger identity management, protection against malicious event injection, and explicit rules about how long location-linked events are retained. These issues are not yet solved by the current prototype, but the prototype helps identify them in concrete terms.

The prototype therefore establishes an implementation baseline for the information-generation mechanism assumed by the cooperative model. Its present scope is limited to sensing, communication, storage, aggregation, and visualisation. Further validation is required before drawing conclusions about real-world performance, including integration of the score-based guidance policy, comparison of BLE-derived events with ground-truth parking behaviour, evaluation of latency and reliability, assessment of mobile-platform and energy constraints, and field testing under realistic operating conditions.

7. Discussion: Bridging Simulation and Deployment

The proposed approach combines a cooperative parking-guidance model with an implementation-oriented data acquisition and aggregation pipeline. This section discusses how these components inform one another and what they imply for deployment. This section therefore focuses on the bridge between them.

7.1. Practical Implications for Incremental Deployment

One of the most encouraging outcomes of the combined study is that the system has a plausible *incremental* deployment path. Many smart-city systems fail because they are only useful after expensive full-scale deployment. Our approach is different. The simulation suggests that moderate participation can already reduce search time. This means that an early-stage deployment could be useful even before ubiquity is reached.

A realistic incremental path might proceed through four phases:

1. **Evidence collection.** Users install the app, pair the in-vehicle beacon, and begin contributing parking-related events.

2. **Descriptive heatmaps.** The system provides map-based awareness of zones with repeated parked or departing events.
3. **Decision support.** The server or app computes the score function from Section 3 and ranks candidate areas.
4. **Advanced optimisation.** Historical trends, confidence estimation, and route-aware suggestions are added on top of the same event pipeline.

The important point is that each phase adds value without requiring the architecture to be redesigned from scratch. This is one of the strongest arguments in favour of the infrastructure-less approach.

7.2. Limitations and Threats to Validity

Despite the positive results, several limitations must be acknowledged.

7.2.1. Prototype Limitations

The current prototype has been validated functionally, not through a longitudinal field trial. The current evaluation does not include real-world measurements of performance metrics such as accuracy, latency, and reliability. The prototype should therefore be interpreted as a proof of concept demonstrating the feasibility of the proposed BLE-based event-driven architecture. BLE disconnection is a useful proxy for parking-related state changes, but it is not perfect ground truth. Beacon placement, smartphone operating-system behaviour, temporary connection loss, and user habits can all affect event reliability. Moreover, the prototype currently emphasises descriptive heatmap visualisation rather than full recommendation generation.

7.2.2. Simulation Limitations

The SUMO environment is synthetic and controlled. This is useful for isolating the behavioural effects of information sharing, but it cannot capture the full heterogeneity of a real city, including mixed traffic behaviour, curbside regulations, temporal variation in parking demand, and adaptation to recommendation systems. The simulation also assumes cooperative agents that use the shared heatmap according to the defined decision model rather than explicitly reasoning about uncertainty, erroneous observations, or adversarial information. Consequently, the quantitative findings should be interpreted as evidence of the behaviour of the proposed guidance model under controlled conditions rather than as direct predictions of performance in a specific real-world city.

Accordingly, the validation performed in this study should be understood as internal and behavioural validation of the model under controlled conditions rather than empirical calibration against a specific urban parking system. The deployment-dependent parameters of the generalised evidence model require calibration using field observations before city-specific quantitative predictions can be made.

7.2.3. Data-Quality and Fairness Issues

Infrastructure-less systems inherit the biases of participation. Areas with more participating users may become overrepresented, while disadvantaged areas or user groups may contribute less data. Likewise, popular recommendations may create self-induced congestion or local overcompetition unless the guidance logic accounts for crowding.

7.2.4. Privacy and Security

A deployable version must address privacy more directly than was necessary in the present prototype. Location data, even when event-driven, are sensitive. Future work should consider pseudonymisation, data minimisation, retention control, user consent management, and robustness against malicious or fabricated reports.

These limitations do not undermine the contribution of the paper. Rather, they frame it correctly. This work establishes a coherent baseline architecture and demonstrates that the basic cooperative idea is both useful and feasible. It does not yet claim fully mature operational deployment.

7.3. Limitations of the Study

To provide a transparent evaluation of the proposed framework, we distinguish between the limits affecting the generalisability of our scientific conclusions and the implementation constraints of the engineering prototype.

7.3.1. Limitations on the External Validity of Scientific Conclusions

While the simulation campaign provides critical insights into system behaviour, certain modelling assumptions constrain its immediate generalisability:

- **Synthetic vs. Real Urban Topographies:** The simulations rely on synthetic road networks. Real-world urban environments feature complex topologies, unpredictable congestion patterns, and highly dynamic parking behaviours.
- **Agent Behaviour Assumptions:** Driver agents are modelled with deterministic preference parameters (e.g., α), whereas real human drivers may exhibit dynamic, irrational, or modelled decision strategies.

7.3.2. Implementation Limitations of the Engineering Prototype

The physical IoT prototype demonstrates proof-of-concept feasibility; however, several engineering constraints remain:

- **Server-Side Integration and Offloading:** The current prototype executes scoring locally; fully offloading scoring function calculations to server-side aggregated data remains to be deployed in production.
- **Scalability and Hardware Bottlenecks:** Processing capacity, bandwidth limits, and stale-data management present performance bottlenecks when scaling edge devices to large-scale deployments.
- **Data Trust and Security:** The current system assumes trusted sensor inputs and lacks dedicated mechanisms against anomalous or malicious data reports.

7.4. Research Agenda and Future Work

These limitations naturally outline the roadmap for future refinements without requiring a change in architectural paradigm:

- **Advanced Agent Modelling:** Integrating frameworks such as AGAMAS to support dynamic agent responses.
- **Privacy and Robustness Engineering:** Implementing explicit uncertainty estimation, stale-data decay, and privacy-preserving retention policies.

8. Conclusions

This article presented an infrastructure-less cooperative parking guidance approach that combines agent-based modelling, heatmap-based information sharing, and an IoT-enabled prototype architecture. The central idea is that participating vehicles can contribute lightweight parking-related evidence without requiring fixed roadside infrastructure. This evidence can then be aggregated into spatial heatmaps and used to support parking decisions that balance parking opportunity against walking distance.

The proposed model formalises the event semantics, heatmap representation, evidence update process, and score-based guidance strategy. The prototype complements this model by showing how parking-related evidence can be generated in practice using a BLE beacon,

a smartphone application, a back-end service, and a database. This implementation path is important because it shows that the information required by cooperative guidance can be collected through commodity devices rather than through dedicated parking sensors, cameras, or smart meters.

The simulation results indicate that cooperative information sharing can reduce parking search time even under partial adoption. They also show that system performance depends on the amount of initial heatmap information and on the preference parameter controlling the trade-off between parking opportunity and walking distance. These findings support the practical relevance of the proposed approach, especially in open urban parking scenarios where real-time information is scarce and infrastructure deployment is costly.

From the simulation, the scientific value of this study lies in answering the research questions posed at the outset regarding system behaviour, adoption dynamics, and preference trade-offs:

- **Addressing RQ1 (User Preferences):** The simulation findings of Section 4.3 demonstrate that tuning user preference parameters effectively manages the trade-off between search time and final walking distance. The simulation results indicate that configuring the parameter to maximise spot availability reduces search time, whereas optimising it for proximity minimises the distance to the destination. Prioritising search time significantly reduces cruising in traffic, albeit at the expense of longer walking distances, proving that customisable guidance policies are crucial for maximising individual driver utility.
- **Addressing RQ2 (Market Penetration and Adoption):** The evaluation in Section 4.4 reveals that cooperative information sharing yields tangible performance improvements even at low market penetration rates. System efficiency scales positively with the percentage of participating vehicles, confirming that complete market coverage is not a strict prerequisite for generating collective benefits. This conclusion comes from the fact that, as shown in Figure 9, the more agents that adopt the system the less their parking search time and the distance to the destination. Indeed, the more the agents collaborate the more the heatmap is updated during daytime, so the information on available spots is more recent and the system can suggest to the user a better parking spot base on their preferences.
- **Addressing RQ3 (Historical Data and Cold-Start):** The results highlight the critical role of pre-existing heatmap information at the start of the search process. A well-seeded initial state drastically reduces search delays and accelerates system convergence, mitigating the cold-start problem inherent to crowdsourced sensing. Historical data help to retrieve the position and the size of parking areas. When an agent parks somewhere, it means that there is a parking spot. When using the whole history of parking events, the map of the city becomes more realistic and helps the system to suggest the right parking spot to the user.

Complementing these simulation-based findings, the proof-of-concept prototype demonstrates that the supporting parking-evidence pipeline can be instantiated using commodity mobile and BLE components without dedicated roadside parking sensors. Its role is limited to implementation feasibility: the prototype shows that parking-related state changes can be acquired, communicated, stored, aggregated, and visualised, but it does not provide field-validated evidence of sensing accuracy, latency, reliability, or deployment readiness. The study therefore provides two complementary forms of evidence with distinct scopes. The simulation provides controlled quantitative evidence addressing RQ1–RQ3, whereas the prototype provides proof-of-concept evidence that the information-generation mechanism assumed by the model is implementable. Future work will integrate

the score-based guidance policy into the live prototype, compare BLE-derived events with ground-truth parking behaviour, evaluate performance under realistic operating conditions, and investigate confidence management, privacy, robustness, and alternative mechanisms for detecting parking-related state changes. Overall, the results support infrastructure-less cooperative parking guidance as a promising coordination approach while identifying the additional empirical and engineering validation required before real-world deployment.

Author Contributions: Conceptualisation, D.D.S., A.F. and F.M.; methodology, D.D.S., A.F. and F.M.; software, D.D.S. and R.T.; validation, D.D.S., A.F. and F.M.; formal analysis, D.D.S., A.F. and F.M.; investigation, D.D.S., A.F. and F.M.; writing—original draft preparation, D.D.S., A.F. and F.M.; writing—review and editing, A.F. and F.M.; supervision, A.F. and F.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Prototype Message Structure and Query Parameters

This appendix summarises the main information exchanged between the mobile application and the back end in the current prototype. The message structure is intentionally compact, since the goal of the system is to support lightweight event reporting and responsive heatmap queries while remaining compatible with a low-cost deployment model based on commodity smartphones and a standard client–server architecture.

Table A1. Representative fields used in the prototype event and query messages.

Context	Field	Meaning
Event upload	Status	Current state inferred by the application, e.g., `parked` or `in motion`
Event upload	Latitude, longitude	Position associated with the detected state transition
Event upload	City, region	Human-readable geographic metadata sent by the application
Server-side enrichment	Timestamp	Time at which the event is recorded and stored
Heatmap request	Start, end	Time interval defining which stored events should be considered
Heatmap request	Centre coordinates	Geographic centre of the currently visible area in the application
Heatmap request	Radius/granularity	Parameters used to delimit the query region and aggregate nearby points
Heatmap response	Position, counter, status	Aggregated point returned to the application for heatmap rendering

In the current prototype, heatmap generation uses a daily temporal window. This choice bounds the amount of evidence considered by each request, reduces visual clutter, and keeps the interaction model simple for exploratory use. The same interface can be extended in later versions with configurable time windows, recency-aware weighting, or confidence annotations.

Appendix A.1. Design Rationale for the Message Schema

The event schema is deliberately minimal. At this stage, the purpose of the prototype is not to model every possible nuance of parking behaviour, but to validate an operational pipeline in which parking-related state changes can be detected, uploaded, stored, and transformed into a spatial visualisation.

A compact schema provides several practical advantages. First, it reduces communication overhead and simplifies implementation on the mobile side. Second, it makes server-side validation and storage more straightforward, since only a small number of fields must be checked and persisted for each event. Third, it keeps the data model interpretable, which is useful during prototype testing and early deployment-oriented experimentation.

Importantly, this minimal design does not preclude richer future behaviour. Historical confidence can be computed from stored timestamps and event density. Recency-sensitive scoring can be introduced without changing the basic event format. Likewise, recommendation logic can evolve above the same persistence layer. The current schema should therefore be understood as a minimal but extensible substrate rather than as a fixed final interface.

Appendix A.2. Possible API Extensions

A natural next step would be to enrich the request and response model so that the prototype can support more selective recommendation policies. For example, query parameters could include user preferences corresponding to the α-style trade-off studied in the simulation, allowing the back end to rank areas differently for drivers who prioritise walking distance, parking likelihood, or search time. Similarly, responses could include freshness or confidence metadata in addition to aggregated points.

Another possible extension concerns the spatial semantics of the query. In the current version, the application requests data for a circular neighbourhood centred on the visible map area. A more advanced implementation could instead query a corridor aligned with the intended route, or return candidate areas ranked by a joint measure of parking opportunity and detour cost. These extensions would enrich the behaviour of the system without requiring a fundamental change to the overall architecture.

References

1. Shen, T.; Hong, Y.; Thompson, M.M.; Liu, J.; Huo, X.; Wu, L. How does parking availability interplay with the land use and affect traffic congestion in urban areas? The case study of Xi'an, China. *Sustain. Cities Soc.* **2020**, *57*, 102126. [CrossRef]
2. Rodríguez, A.; Alonso, B.; Moura, J.L.; dell'Olio, L. Analysis of user behavior in urban parking under different level of information scenarios provided by smart devices or connected cars. *Travel Behav. Soc.* **2024**, *37*, 100847. [CrossRef]
3. Channamallu, S.S.; Kermanshachi, S.; Rosenberger, J.M.; Pamidimukkala, A. A review of smart parking systems. *Transp. Res. Procedia* **2023**, *73*, 289–296. [CrossRef]
4. Lin, T.; Rivano, H.; Mouël, F.L. A survey of smart parking solutions. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 3229–3253. [CrossRef]
5. Biyik, C.; Allam, Z.; Pieri, G.; Moroni, D.; O'Fraifer, M.; O'Connell, E.; Olariu, S.; Khalid, M. Smart Parking Systems: Reviewing the Literature, Architecture and Ways Forward. *Smart Cities* **2021**, *4*, 623–642. [CrossRef]
6. Abu Shroud, M.; Eame, M.; Elsaghayer, E.; Almabrouk, A.; Nassar, Y. Challenges and Opportunities in Smart Parking Sensor Technologies. *Int. J. Electr. Eng. Sustain.* **2023**, *1*, 44–59. [CrossRef]
7. De Soricellis, D.; Ferrando, A.; Muzzini, F. Evaluating Infrastructure-Less Cooperative Parking Guidance via Agent-Based Simulation. In *Proceedings of the International Conference on Practical Applications of Agents and Multi-Agent Systems*; Springer: Cham, Switzerland, 2025; pp. 53–64.
8. Ma, C.; Huang, X.; Li, J. A review of research on urban parking prediction. *J. Traffic Transp. Eng. (Engl. Ed.)* **2024**, *11*, 700–720. [CrossRef]
9. Ayala, D.; Wolfson, O.; Xu, B.; DasGupta, B.; Lin, J. Parking slot assignment games. In Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (GIS '11), Chicago, IL, USA, 1–4 November 2011; pp. 299–308. [CrossRef]

10. Boudali, I.; Ouada, M.B. Smart Parking Reservation System Based on Distributed Multicriteria Approach. *Appl. Artif. Intell.* **2017**, *31*, 518–537. [CrossRef]
11. Murano, A.; Stranieri, S.; Mittelmann, M. Multi-Agent Parking Problem with Sequential Allocation. In *Proceedings of the 15th International Conference on Agents and Artificial Intelligence, ICAART 2023, Lisbon, Portugal, 22–24 February 2023*; Rocha, A.P., Steels, L., van den Herik, H.J., Eds.; SCITEPRESS: Setúbal, Portugal, 2023; Volume 3, pp. 484–492. [CrossRef]
12. Khalidi, N.E.; Benabbou, F.; Sael, N. Distributed parking management architecture based on multi-agent systems. *IAES Int. J. Artif. Intell.* **2021**, *10*, 801–809. [CrossRef]
13. Hassine, S.B.; Kooli, E.; Mraihi, R. Multi-agent Smart Parking System with Dynamic Pricing. In *Proceedings of the 14th International Conference on Soft Computing and Pattern Recognition (SoCPaR 2022)*; Lecture Notes in Networks and Systems; Springer: Cham, Switzerland, 2023; Volume 648, pp. 781–790. [CrossRef]
14. Rizvi, S.R.; Zehra, S.; Olariu, S. MAPark: A Multi-Agent Auction-Based Parking System in Internet of Things. *IEEE Intell. Transp. Syst. Mag.* **2021**, *13*, 104–115. [CrossRef]
15. Icarte-Ahumada, G.; He, Z.; Godoy, V.; García, F.; Oyarzún, M. A Multi-Agent System for Parking Allocation: An Approach to Allocate Parking Spaces. *Electronics* **2025**, *14*, 840. [CrossRef]
16. Dafflon, B.; Contet, J.; Gechter, F.; Gruer, P. Toward a Reactive Agent Based Parking Assistance System. In Proceedings of the 2012 IEEE 24th International Conference on Tools with Artificial Intelligence (ICTAI), Athens, Greece, 7–9 November 2012; pp. 500–507. [CrossRef]
17. Hassoune, K.; Dachry, W.; Moutaouakkil, F.; Medromi, H. Dynamic Parking Guidance Architecture Using Ant Colony Optimization and Multi-agent Systems. *J. Adv. Inf. Technol.* **2020**, *11*, 58–63. [CrossRef]
18. Zhang, X.; Zhao, C.; Liao, F.; Li, X.; Du, Y. Online parking assignment in an environment of partially connected vehicles: A multi-agent deep reinforcement learning approach. *Transp. Res. Part C Emerg. Technol.* **2022**, *138*, 103624. [CrossRef]
19. Stenneth, L.; Wolfson, O.; Xu, B.; Yu, P.S. PhonePark: Street Parking Using Mobile Phones. In *Proceedings of the 13th IEEE International Conference on Mobile Data Management, MDM 2012, Bengaluru, India, 23–26 July 2012*; Aberer, K., Joshi, A., Mukherjea, S., Chakraborty, D., Lu, H., Venkatasubramanian, N., Kanhere, S.S., Eds.; IEEE Computer Society: Los Alamitos, CA, USA, 2012; pp. 278–279. [CrossRef]
20. Nawaz, S.; Efstratiou, C.; Mascolo, C. ParkSense: A smartphone based sensing system for on-street parking. In *Proceedings of the 19th Annual International Conference on Mobile Computing and Networking, MobiCom'13, Miami, FL, USA, 30 September–4 October 2013*; Helal, S., Chandra, R., Kravets, R., Eds.; ACM: New York, NY, USA, 2013; pp. 75–86. [CrossRef]
21. Nandugudi, A.; Ki, T.; Nuessle, C.; Challen, G. PocketParker: Pocketsourcing parking lot availability. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing, UbiComp '14, Seattle, WA, USA, 13–17 September 2014*; Brush, A.J., Friday, A., Kientz, J.A., Scott, J., Song, J., Eds.; ACM: New York, NY, USA, 2014; pp. 963–973. [CrossRef]
22. Attiah, A.; Alahmadi, S.; Hakeem, A.; Mohaisen, L.; Almakky, A.G.; Alhebshi, R.M. FedPark: A Federated Learning Crowdsensing Solution for On-Street Parking Availability. *IEEE Access* **2025**, *13*, 99717–99733. [CrossRef]
23. Elassy, M.; Al-Hattab, M.; Takruri, M.; Badawi, S. Intelligent transportation systems for sustainable smart cities. *Transp. Eng.* **2024**, *16*, 100252. [CrossRef]
24. Soto, I.; Calderón, M.; Amador, O.; Urueña, M. A survey on road safety and traffic efficiency vehicular applications based on C-V2X technologies. *arXiv* **2024**, arXiv:2403.17851. [CrossRef]
25. Darqaoui, M.; Coulibaly, M.; Errami, A. Cellular-V2X and VANET(DSRC) Based End-to-End Guidance for Smart Parking. In *Proceedings of the Wireless Internet—16th EAI International Conference, WiCON 2023, Athens, Greece, 15–16 December 2023, Proceedings*; Maglaras, L.A., Douligeris, C., Eds.; Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering; Springer: Cham, Switzerland, 2023; Volume 527, pp. 3–13. [CrossRef]
26. *ETSI TS 103 916 V2.1.1*; Intelligent Transport Systems (ITS); Parking Availability Service; Release 2; ETSI Technical Specification. European Telecommunications Standards Institute: Sophia Antipolis, France, 2024.
27. Lopez, P.A.; Behrisch, M.; Bieker-Walz, L.; Erdmann, J.; Flötteröd, Y.P.; Hilbrich, R.; Lücken, L.; Rummel, J.; Wagner, P.; Wiessner, E. Microscopic Traffic Simulation using SUMO. In Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA, 4–7 November 2018; pp. 2575–2582. [CrossRef]
28. DLR SUMO Team. LIBTRACI: Library for SUMO Traffic Control Interface. 2025. Available online: https://sumo.dlr.de/docs/Libtraci.html (accessed on 10 March 2025).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.