IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 26 September 2023, accepted 23 October 2023, date of publication 27 October 2023, date of current version 2 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3328233

RESEARCH ARTICLE

SCOPE: Smart Cooperative Parking Environment

MUHAMED ALARBI, (Member, IEEE), ABDELKAREEM JARADAT, (Member, IEEE), HANAN LUTFIYYA, (Senior Member, IEEE), AND ANWAR HAQUE, (Member, IEEE)
Department of Computer Science, The University of Western Ontario (UWO), London, ON N6A 5B7, Canada

Corresponding author: Muhamed Alarbi (malarbi@uwo.ca)

This work was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada. The work of Muhamed Alarbi was supported by a scholarship from the Libyan Ministry of Higher Education and Scientific Research.

ABSTRACT The shortage of parking spaces in metropolitan cities has become a significant challenge, leading to wasted time, money, traffic congestion, and environmental pollution. While smart parking solutions offer potential relief, existing systems often struggle with integration and coordination issues in the complex smart city ecosystem. In response, this paper introduces SCOPE, a cooperative distributed system architecture and interaction model that facilitates the management of parking spaces in a smart city through coordination and autonomous interactions. The system leverages an overlay network, a hierarchical and spatial structure of coordination nodes, and an integration layer to organize traffic and communication among facilities. By incorporating a sharing economy business model, SCOPE maximizes parking resource usage, merges public and private parking resources, and provides economic opportunities for private parking owners. The evaluation results demonstrate that SCOPE significantly reduces search time, traffic, cost, and air pollution while improving driver satisfaction. This novel approach presents a comprehensive solution to the challenges of smart parking management in metropolitan cities, paving the way for more efficient, sustainable, and economically viable urban environments.

INDEX TERMS Smart parking, smart city, smart agents, cooperative distributed systems, cooperative AI, cloud, edge, IoT.

I. INTRODUCTION

As urban populations continue to grow, the number of automobiles in circulation concurrently increases, leading to a shortage of available parking spaces in many cities. This scarcity of parking space results in wasted time, money, and traffic congestion, contributing to environmental pollution. For instance, UK drivers spend an average of 44 hours annually searching for parking spots, with an estimated cost of approximately $23.3 billion per year [1]; nearly one-third of all vehicles on the roads in a given city are searching for parking spots at any given time [2]. As per GeoTab's Urban Infrastructure dataset from 2021, Toronto drivers spent an average of 60 hours seeking parking spaces [3]. Figure 1 visually represents these challenges, with areas in Toronto exhibiting significant parking difficulties highlighted in red. The search for parking index heatmap is derived from two parameters in GeoTab's dataset (SearchingForParkingRatio, AvgTimeToParkRatio) and is scaled from 1 to 10. Higher values correspond to increased driving times dedicated to searching for parking.

Smart city planning involves diverse stakeholders. By utilizing technologies such as IoT devices, software, data, user interfaces, and communication networks, smart city planning aims to create more livable and environmentally sustainable cities. Smart digitization and socioeconomic factors are essential components of this planning process planning [4], [5], [6]. However, in a smart city context, the proliferation of smart parking solutions parking leads to integration problems due to their heterogeneity. Existing parking systems can face coordination and interaction challenges in an open smart city ecosystem with diverse stakeholders and services [7]. While cloud-based parking management has been proposed to address the heterogeneity problem [8], [9], [10], it may not be ideal for situations involving multiple stakeholders with conflicting interests, such as in the case of parking spaces, where each stakeholder aims to maximize their revenue from selling parking spots [11]. In addition, in a large city like Toronto, drivers seeking parking space may need to use multiple mobile apps, such as Honk Mobile,

The associate editor coordinating the review of this manuscript and approving it for publication was Xiwang Dong.

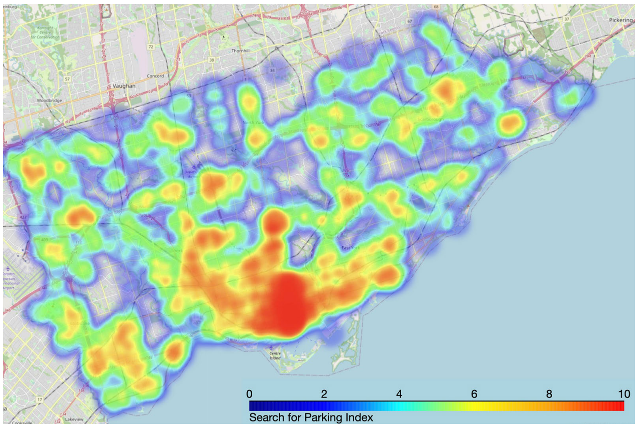

FIGURE 1. Search for Parking Index Heatmap. The index is calculated from two parameters in GeoTab's dataset(SearchingForParkingRatio, AvgTimeToParkRatio). The index is scaled from 1 to 10, with higher values indicating longer driving times spent searching for parking.

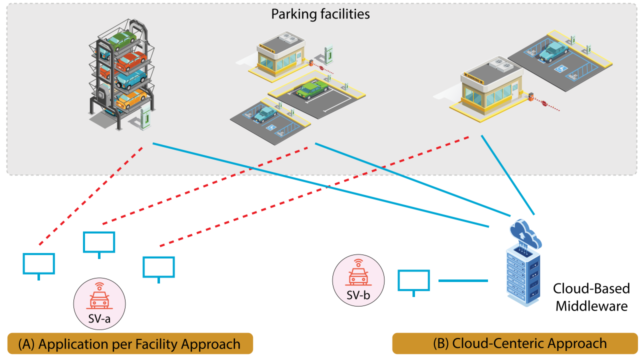

FIGURE 2. Major approaches used in the literature. (A) the application per facility approach, (B) the cloud-centric approach.

SpotHero, BestParking, PARKWHIZ, or Green P, etc., to locate available parking spaces. However, due to a lack of information sharing amongst parking apps, drivers only receive partial information from parking providers, which wastes a valuable and limited resource (i.e., parking space). In order to overcome the challenges associated with smart parking solutions and improve information exchange among diverse stakeholders, an integration layer can be introduced within the smart city ecosystem. This layer serves as a platform that enables seamless communication and cooperation among various parking providers, city authorities, and users. By facilitating the sharing of real-time data on available parking spots, pricing, and occupancy, the integration layer can help optimize parking resources while reducing inefficiencies and redundancies.

Furthermore, the presence of such an integration layer allows for the development of a *sharing economy* business model for parking spaces. The economic benefits of the sharing economy arise from the goal of replacing traditional, centralized services with decentralized yet highly organized (peer-to-peer) networks comprised of active agents [12]. On the one hand, this sharing economy business model maximizes parking resource usage by favouring access over ownership [12]. During business hours, for example, unoccupied private parking spaces in major cities that are accessible to non-owners can significantly reduce the number of parking spaces needing to accommodate other drivers [12], [13]. Further to that, many argue that the sharing economy would aid the socially disadvantaged by providing them with more affordable access to services [14]. The sharing economy, on the other hand, creates opportunities for private parking space owners to earn money by renting out their parking spots during times when they are not in use (such as from 9:00 AM to 5:00 PM).

This work presents Smart Cooperative Parking Environment (SCOPE) that represents a system architecture and interaction model for managing parking space in a smart city through coordination and autonomous interaction using a cooperative distributed systems approach. The contributions of this work are as follows: First, SCOPE creates an overlay network enabling agents to locate and communicate with one another to facilitate interaction. Second, it introduces a hierarchical and spatial structure of coordination nodes that harnesses the Cloud-Edge continuum and 5G networks to organize traffic and communication among agents, reducing network congestion and accelerating processing rate. Third, SCOPE presents an integration layer facilitating autonomous interaction among parking agents in a smart city. Moreover, SCOPE supports a sharing economy business model, merging private and public parking resources and allowing private parking owners to generate passive income from unused parking spaces. Lastly, SCOPE considers driver preferences when serving requests, reducing search time, traffic congestion, and harmful emissions.

The rest of the paper is organized as the following. Section II explains related published work. Section III describes the System design. Section IV describes the formal model. Section V discusses the model architecture. Section VI elaborates on the performance evaluation plan of the proposed architecture. Section VII discusses limitations and real-world deployment. And finally, Section VIII concludes the paper.

II. RELATED WORK

The integration of IoT and cloud computing has led to the development of Smart Parking Systems, resulting in reducing traffic congestion, driving cost, and environmental pollution [15]. Various models of parking systems have been presented in the literature, including IoT-Cloud Smart Parking, Intelligent Smart Parking, Multi-Agent Smart Parking, and IoT-Fog-Cloud Smart Parking [16]. Figure 2 illustrates two major approaches to smart parking systems. The first approach is called **application per facility**, in which a client system communicates with individual parking systems in the area of interest to query available spots for a specified time frame. The second approach is called the **cloud-centric** approach, in which the client system communicates with a centralized system that manages parking facilities. The status of the spots is sent to a central point that handles all booking operations.

A. IoT-CLOUD SMART PARKING

The development of cloud-based smart parking systems has revolutionized the way drivers find parking spaces around the city. These systems rely upon the use of IoT sensors, RFID devices, or satellite cameras to monitor the status of parking spots. The information collected is sent to a cloud-based resource allocator that handles booking and payments. Communication within this type of system is facilitated through the Internet. The main focus of the IoT-Cloud Smart systems is on efficiency in finding parking spots, and its main distinguishing factor is its real-time updates. Melnyk et al. [17] proposed the Smart Parking Management System (SPMS) that helps drivers find available parking slots in multistory parking facilities by providing real-time updates on parking slot status through a mobile app. Hainalkar and Vanjale [18] proposed a system that allows drivers to book specific parking spots and facilitates automatic cashless billing and post-trip booking. Also, the system provides transportation authorities with status updates on each parking area, which helps control urban traffic. Griggs et al. [13] addressed the issue of efficient parking spot utilization by offering a scheme that allows university visitors to use residential parking spaces during daytime hours. The authors study an alternative to the first-come-first-served model. They also focus on determining the appropriate size of parking reserve the university allocates to resolve conflicts in residential parking spots. Pham et al. [19] proposed a parking system in which the cost of parking is determined by the distance and the number of available parking spaces, using wireless sensor networks and RFID technologies. Baranwal et al. [20] tackle the issue of parking in cities by proposing a personalized Parking Recommender System that takes into account multiple quality parameters related to parking, such as walking distance, pricing, and safety. They utilize fuzzy logic to handle the uncertainty in human decision-making.

B. INTELLIGENT SMART PARKING

Intelligent smart parking systems have become popular as they allow drivers to reserve parking spaces in advance based on their preferences and location. These preferences may include factors such as the duration of the stay, previous parking history, and traffic congestion levels. The Intelligent Parking system is also cloud-based but has a unique feature of centralized resource allocation that considers driver preferences. Like IoT-Cloud, it uses the Internet for communication and relies on IoT sensors, RFID, satellite, and cameras for information. Parking Spot status is updated in real-time. The pricing model is dynamic and varies based on the parking lot's capacity. Kanteti et al. [21] proposed an algorithm that allocates parking spaces according to the driver's preferences and provides guidance on the reserved parking spot. Additionally, the pricing algorithm calculates parking fees based on allocation preferences and demand factors. Delot et al. [22] presented a reservation protocol based on event sharing with a mobile P2P architecture, where events represent messages disseminated to notify drivers of available parking spaces in their vicinity. This solution allocates parking spaces in short-range vehicular ad-hoc networks, avoiding competition among vehicles but limited to short-range communication. In contrast, Kotb et al. [23] proposed a Mixed Integer Linear Programming model that minimizes parking cost and search time. Lin et al. [24] introduces a smart parking allocation algorithm (SPA) designed to maximize the benefits generated by a given parking lot while ensuring high-quality parking services. The proposed SPA algorithm predicts driver behaviour and estimated parking traffic based on historical parking records, allowing for improved matching of parking demands with available resources. P. Zhao et al. [25] tackles the issue of effectively managing shared parking resources while guaranteeing parking availability for private space owners (O-users). The authors present a management framework that considers both time and spatial aspects of shared parking that is used to create an intelligent parking management system (IPMS) to model shared parking operations. Babic et al. [10] assessed three parking policies using real-world data and simulations with the aim of fulfilling electric vehicle charging requirements and optimizing revenue from parking resource management.

C. MULTI-AGENT SMART PARKING

Multi-Agent Smart Parking systems use a multi-agent software design paradigm to model the parking environment. Unlike the previous systems, it uses a decentralized resource allocator. Communication among the agents is facilitated through VANET (Vehicular Ad-hoc Networks). The system also relies on IoT sensors, RFID, satellite, and cameras for information. Real-time updates on spot status are provided. The pricing model can be either fixed or dynamic. The Multi-Agent system focuses on agent-based allocation of parking spots, and its primary distinguishing factor is that system functionality is delegated to different agents. Dargaye et al. [26] proposed an agent-based smart parking model. This system recommends parking spots to drivers based on several factors, including the driver's location, speed, and time needed to reach the final destination. The system's functionality is delegated to seven agents: speed, GIS, parking, mobile, routing, and analytics. This system was influenced by the Agent-based Intelligent Parking and Guidance System, which uses mobile technology and a multi-agent system approach to select parking spots based on the desired location and guidance system [27], [28]. Another example is the agent-based parking management system presented by Rizvi et al. [8], which offers drivers the best available parking place based on their preferences.

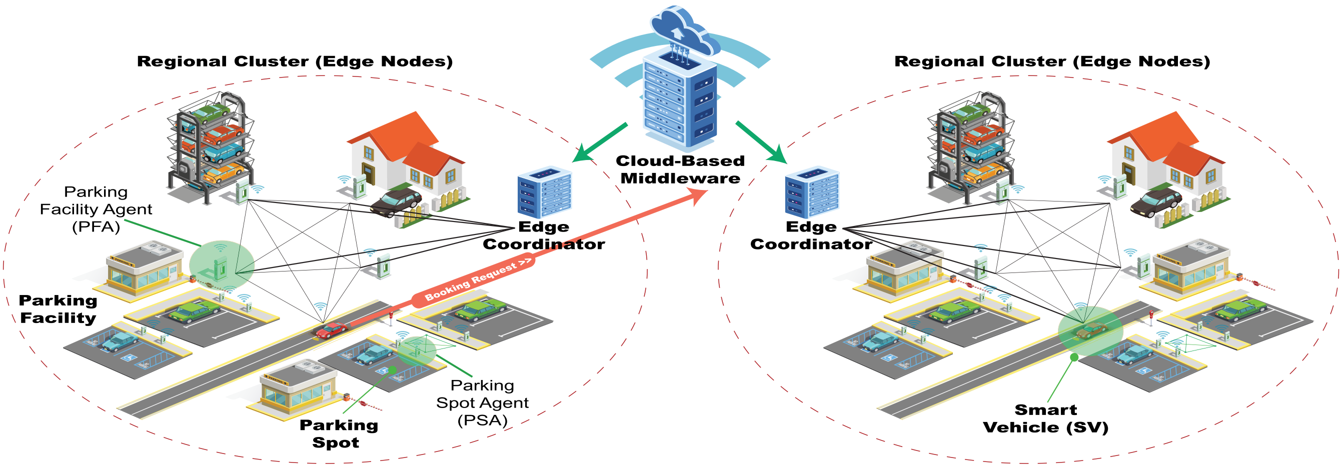

FIGURE 3. High-level architecture of SCOPE.

These preferences include the maximum amount the driver is willing to pay, the parking type, and the acceptable walking distance to the driver's ultimate destination. The system uses a parking broker, a software agent that collects useful route planning information from other traffic intelligence services to increase road safety by reducing the need for drivers to engage with their cellphones or onboard devices. The Multi-Agent Auction-based Parking (MAPark) system proposed is designed to benefit both drivers and Parking Facility Providers (PFPs) by utilizing real-time demand-based pricing for parking places [9].

Although the aforementioned systems leverage the multi-agent paradigm to model the parking environment and display elements of the sharing economy, their approach is rooted in functional modelling, where each agent is assigned a specific task. On the other hand, SCOPE significantly diverges from this structure and functionality by recognizing the multi-stakeholder nature of the environment.

D. IoT-FOG-CLOUD SMART PARKING

The emergence of Fog Computing has caught the attention of researchers exploring the potential of fog computing architecture to solve smart parking management issues. Fog computing architecture provides cloud services at the network's edge, distributing workloads across multiple micro data centers and reducing response time. This makes it an excellent choice for time-sensitive applications like parking management systems. The IoT-Fog-Cloud system uses a multi-tier compute and network fabric architecture. It has a centralized resource allocator and uses LoRaWAN-5G and the Internet for communication among different system components. Information is sourced from IoT sensors, RFID, satellite, and cameras. Spot status updates are provided in real-time, and the pricing model is dynamic. The system focuses on time-sensitive spot updates, and its main distinguishing factor is its reduced response time due to the multi-tier architecture. Hsieh et al. [29] presented a fog computing architecture for managing parking infrastructure in urban cities. The system involves multiple fog nodes that manage parking spots and respond to driver requests, assisted by Roadside Units. Kim et al. presented a similar fog architecture, but it considers private parking lots like those found at restaurants, stores, and bookstores. Tang et al. [30] presented a multi-layer architecture that combines edge nodes in the parking with VANET to get real-time parking availability information and handle parking requests from approaching vehicles. Tondon et al. [31] proposed an online reservation system that uses bookings to reduce driving time, fuel consumption, $CO2$ emissions, and traffic congestion. The system uses cloud and fog computing to track parking space status. Parking facilities report spot status to nearby fog nodes that convey these updates to Roadside Units and cloud computing nodes. Awaisi et al. [32] presented a fog-based smart parking architecture that utilizes computer vision to efficiently identify vacant parking spaces, reducing both time and fuel consumption for drivers.

E. RESEARCH GAP AND CONTRIBUTIONS

The current state of smart parking applications predominantly follows a methodological individualism approach that has persisted for decades, leading drivers in megacities to rely on multiple mobile apps to locate parking. This individualistic approach, once reasonable, has become increasingly inadequate with the emergence of smart city concepts and the rapid, intricate interactions between various smart systems and individuals. Consequently, the need for cooperative solutions has grown [33]. Existing smart parking solutions suffer from a lack of information sharing among apps, which results in drivers receiving only a partial view of available parking spaces and squandering valuable and limited resources like parking spots. Furthermore, the absence of cooperation between parking facilities hinders their ability to adjust pricing strategies and maximize profits. Rather than using a cloud-centric resource management approach or numerous

TABLE 1. Comparison of smart parking systems in the literature.

Ref	System Model	Parking Type	Business Model	Autonomous Interaction	Localized Processing
[17]	Centralized	Public	Facility-App	×	×
[18]	Centralized	Public	Marketplace	×	×
[13]	Centralized	Public/Private	Facility-App	×	×
[19]	Centralized	Public	Marketplace	×	×
[20]	Centralized	Public	Marketplace	×	×
[21]	Centralized	Public	Facility-App	×	×
[22]	Decentralized	Public	Marketplace	×	✓
[23]	Semi-Distributed	Public	Marketplace	×	✓
[24]	Centralized	Public	Marketplace	×	×
[25]	Centralized	Public/Private	Facility-App	×	×
[10]	Centralized	Electric Vehicle	Facility-App	×	×
[26]	Distributed	Public	Marketplace	×	×
[27]	Distributed	Public	Marketplace	×	×
[8]	Centralized	Public	Marketplace	×	×
[9]	Distributed	Public	Procurement	×	×
[29]	Distributed	Public	Marketplace	×	×
[30]	Centralized	Public	Marketplace	×	✓
[31]	Centralized	Public	Marketplace	×	✓
[32]	Distributed	Public	Marketplace	×	×
SCOPE	Cooperative Distributed	Public/Private	Procurement	✓	✓

collaborative agents to assist drivers in identifying suitable parking space, as proposed in [8], [9], and [26], SCOPE portrays parking facilities as independent, smart agents, each holding total control over its parking space. These facilities distribute information in a competitive manner, fostering an environment of trading competition facilitated by a cloud-edge architecture. This highlights a shift from the other systems' collaborative, cloud-centric approach towards a decentralized, competitive model that emphasizes the autonomy of individual parking facilities while incorporating cooperative AI strategies. Table 1 offers a detailed analysis of various smart parking systems discussed in the literature. Each row represents a different system or research paper, identified by its reference. The table compares these systems based on five key dimensions: System Model, Parking Type, Business Model, Autonomous Interaction, and Localized Processing. The System Model describes the architecture. Parking Type indicates whether the system is designed for public, private, or specialized parking. Business Model outlines the operational framework, such as Facility-App or Marketplace. Autonomous Interaction reveals if the system allows for self-governing interactions between parking facilities. Lastly, Localized Processing specifies if data processing occurs locally or is sent to a central server. This table serves as a comprehensive tool for understanding the nuances between SCOPE and the reviewed parking systems.

To address these limitations, the proposed work introduces a novel system architecture that leverages a cooperative approach among public and private parking facilities, such as those found in restaurants, stores, bookstores, and residential areas, with the aim of increasing parking capacity. The proposed system architecture, SCOPE, sets itself apart from existing work by harnessing joint behaviour among parking facilities to augment capacity and streamline the parking experience. SCOPE features a layered architecture and integration framework that establishes a common parking market, enabling public and private parking facilities to engage in trading competition to meet drivers' parking spot requests. This benefits both drivers and parking facilities, as it allows parking providers to adjust their pricing based on demand, optimizing profit generation.

III. SYSTEM DESIGN

In this section, we highlight the fundamental concepts driving the design of SCOPE.

A. INTERACTION MODEL

SCOPE models the smart parking environment as an assembly of autonomous economic agents, which are Parking Facilities and Smart Vehicle agents, alongside hierarchical coordination agents called Edge Coordinators and Cloud-based Middleware. The implementation of SCOPE agents follows the CIR-Agent architecture [34]. This means that every agent makes rational decisions and executes tasks independently. Furthermore, coordination agents facilitate negotiation and resolve knowledge interdependence among parking facility agents. Figure 3 graphically illustrates SCOPE agents' hierarchical and spatial structure. Using autonomous interaction, SCOPE enables collaboration among diverse parking facilities, allowing parking agents to address challenges beyond individual capabilities jointly. One such challenge is acquiring comprehensive knowledge about available parking spaces in an Area of Interest (AoI) and utilizing this information to fulfill parking requests efficiently. Another advantage of SCOPE's autonomous interaction is its ability to handle conflicts in a parking facility agent's booking schedule. Such conflicts may arise from parking booking violations, where a car remains in a parking spot beyond its allotted time slot. If the parking facility is at total capacity, the parking facility agent cannot redirect the approaching vehicle to another parking spot. In this situation, neighbouring parking agents can offer extra parking spaces to resolve the conflict. This interaction can also be triggered when a driver approaches a fully occupied parking facility without a booking. In this scenario, since SCOPE is unaware of the driver's preferences, the parking facility agent asks neighbouring parking facility agents to submit their offers and directs the driver to a neighbouring parking facility that provides the best commission to the parking agent initiating the interaction. These collaborative efforts encourage parking facility agents to willingly share their resources, fostering a more efficient and cooperative smart parking environment.

B. SMART AGENTS

SCOPE consists of the following agents:

Smart Vehicle (SV): An interface for drivers to interact with SCOPE through web applications, mobile applications, or smart vehicle gadgets.

Parking Spot Agent (PSA): A PSA is a combination of a computing unit and a sensor for monitoring a parking spot's status. A parking lot contains multiple PSAs, each implementing the CIR-Agent architecture.

Parking Facility Agent (PFA): A PFA is a CIR-Agent deployed at a public or private parking facility and responsible for managing the parking space resources. A PFA communicates with other PFAs in the same geographical cluster through an Edge Coordinator (EC).

The Edge Coordinator (EC): An EC is an agent that organizes and facilitates communication between PFAs within a given geographic cluster. ECs can also talk to their neighbours to manage drivers' preferences across multiple clusters. In addition, an EC acts as an abstraction layer enabling SVs to request directly from roadside nodes.

Cloud-Based Middleware (CBM): The CBM is the initial point of contact for SVs seeking parking spaces through SCOPE. CBM's abstraction layer hides the heterogeneity of the parking system and serves as a client-facing API gateway, decoupling SVs from the underlying parking infrastructure. SV requests are forwarded to the CBM's findParkingSpot API, which redirects them to PFAs within the area of interest (AOI). PFAs then submit bids to accommodate the request. Additionally, CBM functions as a global coordination node for regional EC nodes. Figure 6 illustrates the multi-layer architecture of SCOPE.

C. SINGLE-ROUND, SEALED-BID, REVERSE AUCTION STRATEGY

The interest in employing auction-based methods to address resource allocation and pricing challenges within the sharing economy has been steadily increasing [35]. As a decentralized system, SCOPE lacks a global view of available parking spaces. To address this, Edge Coordinators (ECs) act as auctioneers, conducting Single-Round, Sealed-Bid, Reverse Auctions [36] to resolve knowledge interdependencies among Parking Facility Agents (PFAs). In sealed-bid auctions, participants submit bids without knowing offers from other participants. The reverse auction model encourages suppliers to lower resource prices or service costs to secure clients' business, with buyers and sellers exchanging roles. This approach drives prices downward and can result in savings of 10-40%. [37]. Resolving knowledge interdependency process is formally defined in Section III.

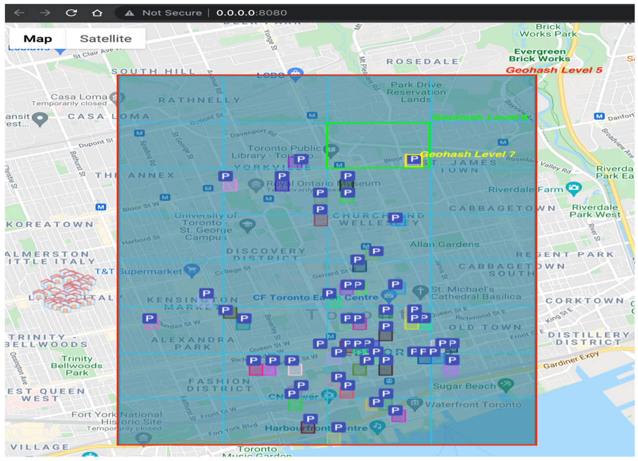

FIGURE 4. Distribution of parking facilities across geographical clusters.

D. GEOGRAPHICAL CLUSTERING

Geohash [38] is a hierarchical geocoding system used for spatial indexing [39]. It encodes location coordinates and groups nearby points on the globe with varying resolutions across 12 levels, with cell sizes decreasing as levels increase. SCOPE employs a hierarchical, spatial structure of coordination nodes, utilizing the Cloud-Edge continuum and 5G networks to manage traffic and communication among agents. PFAs are grouped into geographical clusters based on their Geohash level 6 code, which covers an area of 1.2 km x 609.4M. Each geographical cluster represents a Geohash level 6 area, and an Edge Coordinator (EC) is assigned to facilitate interaction among PFAs. Figure 4 depicts the Geographical clustering.

IV. SCOPE FORMULATION

This section presents the mathematical model of SCOPE agents and the interaction model. In addition, a formal definition of the parking spot selection algorithm is presented. Table 2 provides frequently used variables and their description.

TABLE 2. Frequently used variables.

Variables	Description
CBM	Cloud Based Middleware.
EC	Edge Coordinator.
PFA	Parking Facility Agent.
PSA	Parking Spot Agent.
SV	Smart Vehicle.
$\mathbb{M}$	set of CBMs.
$\mathbb{E}^m$	set of ECs connected to a CBM.
$\mathbb{F}^e$	set of PFAs connected to an EC.
$\mathbb{S}^f$	set of PSAs within a PFA.
$\mathbb{V}^f$	set of SVs.
Ψ	set of all possible parking preferences.
Ψ^c	a subset of Ψ characterized as cost preferences.
Ψ^b	a subset of Ψ characterized as benefit preferences.
Ψ_v	a subset of Ψ represents driver-selected parking preferences.
$\Delta_{\mathcal{H}}$	HTTP Message.
$\Delta_{\mathcal{P}}$	Publish-Subscribe Message.
π_{find}	find parking spot request.
$\pi_{forward}$	CBM forwarding π_{find} to and EC in Area of Interest.
$\pi_{publish}$	EC's announcement to PFAs to send their proposals.
π_{bid}	parking facility bid in response to $\pi_{publish}$ message.
ϕ	potential parking spot that satisfies the driver criteria.
Φ	set of parking proposals (alternatives)
$\mathbb{F}^e_*$	set of PFAs that submitted bids .
ω^b	the most significant criterion in driver preferences.
ω^w	the least significant criterion in driver preferences.
α_i^b	a coefficient proportional to the relative importance between ω^b and ω_i.
β_i^w	a coefficient proportional to the relative importance between ω_i and ω^w.
ξ_L	an indicator of the consistency of the criteria pairwise comparisons.
$\mathbf{W}$	weights vector for driver preferences.
Θ	a score vector representing a ranking of PFA bids based on driver preferences.
$\bar{\Phi}$	normalized alternatives matrix.
θ^*	score of winning bid.
$\pi_{confirm}$	a request for confirmation sent to the PFA that won the auction.
π_{ack}	acknowledgment message to state the commitment.
$\pi_{booking\_ack}$	booking confirmation message to the SV

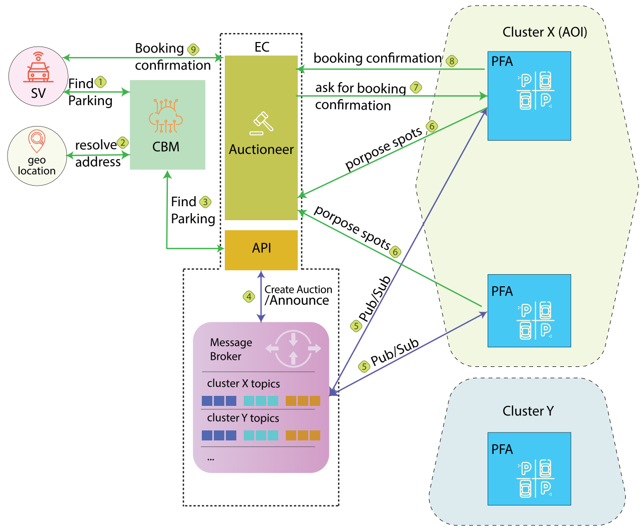

FIGURE 5. The interaction model for the find parking spot (FPS) scenario.

A. DEFINITIONS

SCOPE contains a set of CBMs defined as $\mathbb{M}$, such that:

$$\mathbb{M} = \{m_0, \ldots, m_K\} \tag{1}$$

where K is the number of CBMs in SCOPE, and each CBM $m \in \mathbb{M}$ represents a cloud-based middleware that manages the interactions within the city. A city is divided into multiple geographical clusters, where every geographical cluster is managed by an EC. Therefore, a CBM, m, manages a set of ECs, $\mathbb{E}^m$, such that:

$$\mathbb{E}^m = \{e_0, \ldots, e_{L^m}\} \tag{2}$$

where L^m is the number of ECs within a CBM, m, and each EC, $e \in \mathbb{E}^m$ is the edge coordinator which facilitates the interaction among parking facilities within the geographical cluster that belongs to the CBM, m. A geographical cluster contains multiple parking facilities. A set of PFAs, $\mathbb{F}^e$, is defined as follows:

$$\mathbb{F}^e = \{f_0, \ldots, f_{N^e}\} \tag{3}$$

such that N^e is the number of PFAs in e. A parking facility contains multiple parking spots. A set of PSAs $\mathbb{S}^f$ is defined as follows:

$$\mathbb{S}^f = \{s_0, \ldots, s_{P^f}\} \tag{4}$$

such that P^f is the number of PSAs in f. A set of SVs, $\mathbb{V}$, is defined as:

$$\mathbb{V} = \{v_0, \ldots, v_G\} \tag{5}$$

where G represents the number of all smart vehicles in the city.

B. FIND PARKING SPOT SCENARIO

Finding a parking spot starts when a smart vehicle sends a request to SCOPE. The request is sent from the SV, v, to the CBM, m, with a set of driver preferences. This is defined as:

$$\pi_{find} = \Delta_{\mathcal{H}}(v, m, \Psi_v) \tag{6}$$

SCOPE maintains a set of all possible parking preferences, Ψ, divided into the set of cost preferences, Ψ^c, and the set of benefit preferences, Ψ^b. The price and walking distance benefit are examples of cost preferences where smaller values are preferred, while parking ratings are an example of benefit preferences where larger values are preferred. This is defined as follows:

$$\Psi = \Psi^c \cup \Psi^b = \{(\psi_0, \lambda_0), \ldots, (\psi_i, \lambda_i), \ldots\} \tag{7}$$

where ψ_i represents the criterion, and λ_i is a binary indicator that determines whether ψ_i is a cost or benefit criterion.

The driver's preferences set is considered as a set of criterion, Ψ_v, of size, Z. Each criterion bounds the preferred values provided to the CBM, m. Each criterion is bounded by a lower limit and an upper threshold such that:

$$\Psi_v = \{\psi_0, \ldots, \psi_Z\} \tag{8}$$

$$\psi_i \in \Psi_v = (\psi_i^l, \psi_i^u) \tag{9}$$

$$\Psi_v \subseteq \Psi \tag{10}$$

where $\psi_i \in \Psi_v$ is a single criterion with the lower limit, ψ_i^l, and the upper threshold, ψ_i^u. The CBM, m, forwards the request, π_{find}, to the EC, e, associated with the geographical cluster that matches the preferred area of interest. This forwarding is defined as:

$$\pi_{forward} = \Delta_{\mathcal{H}}(m, e, \pi_{find}) \tag{11}$$

The EC, e, then broadcasts this request to all subscribed PFAs, $\mathbb{F}^e$, and forwards the criteria set to the subscribers as follows:

$$\pi_{publish} = \Delta_{\mathcal{P}}(e, f, \pi_{forward}), \quad \forall f \in \mathbb{F}^e \tag{12}$$

Each PFA evaluates the requested preferences and replies back to the CBM with a set of bids that represent the available matching spots to the criteria set:

$$\pi_{bid} = \Delta_{\mathcal{H}}(f, e, \Phi_f), \quad f \in \mathbb{F}^e \tag{13}$$

where π_{bid} is the response message from f to e, that contains a list of bids for the matching parking spots in the facility, f. After receiving the bid, SCOPE agent that holds the auction, in this case EC, creates an alternative, ϕ, e.g., a potential parking spot that satisfies the driver criteria. This is defined as follows:

$$\Phi = \{\phi_0, \ldots, \phi_Y\} \tag{14}$$

$$\phi \in \Phi = \phi_f \cup \phi_e \tag{15}$$

$$Z = \|\Psi_v\| = \left\|\phi_f\right\| + \|\phi_e\| \tag{16}$$

where ϕ_f represents a set of criterion values provided by the PFA, f, (e.g., price, spot type, parking type, etc), while ϕ_e represents a set of criterion values provided by the EC, e, (e.g., walking distance to destination, parking ratings, etc.). The size of the set of alternatives is represented by Y. The alternatives set, Φ, is defined as a 2D matrix as:

$$\Phi_{Z\times Y} = \begin{array}{c c} & \begin{array}{cccc} \boldsymbol{\phi_1} & \boldsymbol{\phi_2} & \cdots & \boldsymbol{\phi_Y} \end{array} \\ \begin{array}{c} \boldsymbol{\psi_1} \\ \boldsymbol{\psi_2} \\ \vdots \\ \boldsymbol{\psi_Z} \end{array} & \begin{bmatrix} \phi_1^1 & \phi_2^1 & \cdots & \phi_Y^1 \\ \phi_1^2 & \phi_2^2 & \cdots & \phi_Y^2 \\ \vdots & \vdots & \phi_i^j & \vdots \\ \phi_1^Z & \phi_2^Z & \cdots & \phi_Y^Z \end{bmatrix} \end{array} \tag{17}$$

$$Y = \sum_{f_*}^{f_* \in \mathbb{F}^{e_*}} \left\| \mathbb{S}_*^{f_*} \right\| \leq P^f.N^e \tag{18}$$

such that:

$$f_* \in \mathbb{F}_*^e, \quad f \in \mathbb{F}^e, \quad \mathbb{F}_*^e \subseteq \mathbb{F}^e$$
$$s_* \in \mathbb{S}_*^f, \quad \mathbb{S}_*^f \subseteq \mathbb{S}^f$$

where $\mathbb{F}_*^e$ represents the set of PFAs that submitted bids, $\mathbb{S}_*^f$ represents the set of proposed spots. ϕ_i^j is the i^{th} alternative value (e.g., PFA bid) of the j^{th} criterion, s_* is a parking spot that satisfies all the criteria and is located in the facility, f_*. The number of bids, Y, is defined as the summation of the number of proposed spots, $\mathbb{S}_*^f$, for all proposing facilities, $\mathbb{F}_*^e$.

The alternatives set received by the EC, e, must satisfy the bound conditions such that:

$$\psi_j^l \leq \phi_i^j \leq \psi_j^u$$
$$\phi_i^j \in \Phi \quad , \quad i < Y \quad , \quad j < Z \tag{19}$$

C. PARKING SPOT SELECTION

The Best-Worst Method (BWM) [40] is used to calculate the optimal weights of the user-defined criteria. The weights are used to select the best suitable alternative (e.g., PFA bid) that satisfies the user criteria. In BWM, the driver selects the most important criterion, ω^b, and the least important criterion, ω^w. Afterward, the driver establishes a pairwise comparison between ω^b and the rest of the criteria, $\omega_i \in W$ such that:

$$\omega^b = \alpha_i^b \omega_i \tag{20}$$

where α_i^b is a coefficient proportional to the relative importance between ω^b and ω_i, in other words, α_i^b shows how much the driver prefers criterion ω_b over criterion ω_i. In addition, the driver establishes a pairwise comparison between ω^w and the rest of the criteria, $\omega_i \in W$ such that:

$$\omega^w = \beta_i^w \omega_i \tag{21}$$

where β_i^w is a coefficient proportional to the relative importance between ω_i and ω^w, the following Linear Integer model is used to derive the optimal weights:

$$\min \quad \xi_L \tag{22}$$

$$\text{s.t.} \quad |\omega^b - \alpha_i^b \omega_i| \leq \xi_L \quad \forall i, i < Z \tag{23}$$

$$|\beta_i^w \omega_i - \omega^w| \leq \xi_L \quad \forall i, i < Z \tag{24}$$

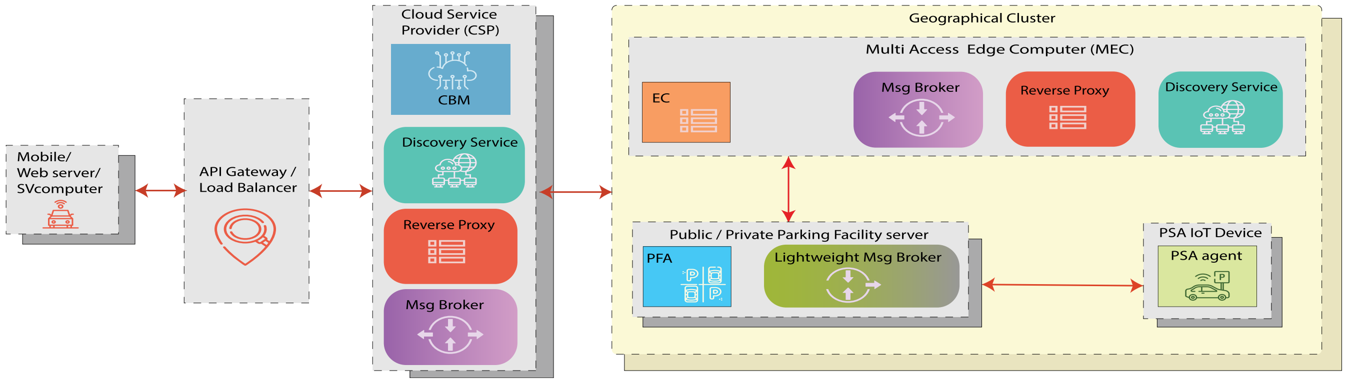

FIGURE 6. **SCOPE multi-layer architecture.**

$$0 \leq \xi_L \tag{25}$$

$$0 \leq \omega_i \leq 1 \quad \forall i, i < Z \tag{26}$$

$$\sum_{i=0}^{i<Z} \omega_i = 1 \tag{27}$$

where ξ_L is considered an indicator of the consistency of the criteria pairwise comparisons. In this context, consistency represents relevance in the driver pairwise comparison for (ω_i, ω_b) and (ω_i, ω_w). The lower values of ξ_L represent higher consistency. By solving the optimization problem, a weights vector **W** is defined as:

$$W = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \dots \\ \omega_Z \end{bmatrix}$$

such that the value of ω_i corresponds proportionally to the importance of the criterion ψ_i. The weights obtained from BWM are used to select the best parking spot based on the driver's preferences. To achieve this, a BWM score vector Θ is constructed. The score vector represents a ranking of PFA bids based on driver preferences. The score vector Θ is defined as:

$$\Theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \dots \\ \theta_Y \end{bmatrix} = \overline{\Phi}^T . W \tag{28}$$

Such that:

$$\theta_j = \sum_{i=1}^{i\leq Z} \omega_i . \overline{\phi}_i^j \quad , \quad \forall j < Y \tag{29}$$

where $\overline{\Phi}$ is the normalized alternatives matrix $\overline{\Phi}_{Z\times Y}$, where each normalized element is defined as:

$$\overline{\phi}_i^j \in \overline{\Phi}_{Z\times Y} = \begin{cases} \dfrac{\psi_i^u - \phi_i^j}{\psi_i^u - \psi_i^l} & , \psi_i \in \Psi_v^c \\ \dfrac{\phi_i^j - \psi_i^l}{\psi_i^u - \psi_i^l} & , \psi_i \in \Psi_v^b \end{cases} \tag{30}$$

The scores vector Θ is sorted in descending order. The score, θ^*, that wins the auction is the maximum score:

$$\theta^* = MAX(\Theta) \tag{31}$$

and the corresponding alternative to the winning score is ϕ^*. The EC sends a confirmation message to the PFA that won the auction i.e., proposed ϕ^*, as follows:

$$\pi_{confirm} = \Delta_{\mathcal{H}}(e, f, \phi^*) \tag{32}$$

The PFA should reply back with an acknowledgment message to state the commitment to the proposed spot:

$$\pi_{ack} = \Delta_{\mathcal{H}}(f, e, \phi^*) \tag{33}$$

Upon receiving the acknowledgment by the EC, the EC sends a booking confirmation message to the SV:

$$\pi_{booking\_ack} = \Delta_{\mathcal{H}}(e, v, \{\phi^*, f, s\}) \tag{34}$$

Suppose the EC did not receive the acknowledgment within a specific time frame. In that case, the winning bid, ϕ^*, is cancelled, the second next maximum score is considered as winning, and the confirmation process is performed again. Figure 5 illustrates how SCOPE agents handle a find parking spot request.

V. THE ARCHITECTURE

In this section, we highlight the main building blocks of SCOPE architecture.

A. ECONOMIC AGENTS

With SCOPE, we envision PFAs as economic agents that join the SCOPE environment for their benefit. SCOPE provides the architecture and framework enabling interaction among these agents. Influenced by the CIR-Agent model [34], SCOPE's PFAs comprise the following modules: Problem Solver, Knowledge Base, Interaction, and Communication.

B. COORDINATION AGENTS

In SCOPE, CBM and EC are coordination agents that enable autonomous interaction and communication among SCOPE's economic agents. In addition to the base smart agent components highlighted in Economic Agent Architecture, Coordination agents have additional components such as Discovery Service, Reverse Proxy, and Message Broker.

C. COMMUNICATION COMPONENTS

SCOPE is designed to operate efficiently within an open environment, where numerous parking agents can coexist without limitations or reliance on each other. In this setting, parking agents have the freedom to enter or exit the environment as they please. Consequently, they do not possess information about other agents' network addresses, such as IP addresses or port numbers. To support seamless interaction in this open environment, SCOPE incorporates the following communication components:

API Gateway/Load Balancer: To ensure scalability and high availability, SCOPE uses multiple CBM instances. SV requests are routed via the API Gateway and Load Balancer, which creates a single-entry point for all clients, abstracting the CBMs layer for the SVs.

Discovery Service: SCOPE employs a multi-replica discovery service that maintains a network registry of available agents' (hostname, address) pairs. This service dynamically discovers an agent's network location (IP address and port) to enable agent communication. Agents continuously report their availability to the discovery service using heartbeats, and their addresses and online statuses are replicated across other discovery services. These services collectively form a peer-to-peer network responsible for managing knowledge about agent addresses. In SCOPE, the Discovery Service is integrated as a component within the Cloud-Based Middleware (CBM) and Edge Coordinators (ECs).

Reverse Proxy Service: SCOPE uses a Reverse Proxy service for dynamic request routing among agents, supporting scalability and high availability. When an agent sends a request, it submits the message and the receiving agent's name to the nearest Edge Coordinator's Reverse Proxy service. The Reverse Proxy queries the Discovery Service registry for the receiving agent's address and uses a load-balancing algorithm to route the request to one of the available instances. The response returns to the sender agent via Reverse Proxy, enabling smooth communication without requiring address lookups by the sending agent. In SCOPE, the Reverse Proxy Service is a Cloud-Based Middleware (CBM) and Edge Coordinators (ECs) component.

Message Broker: The Message Broker enables indirect communication among agents in SCOPE. It allows an agent to send a find parking spot request to nearby agents without direct contact. Using the Publish-Subscribe messaging pattern, agents send messages to the Message Broker, which publishes the messages to potential recipients. This bidirectional messaging approach occurs through an intermediary system, the Message Broker, without direct communication between data publishers and consumers. In SCOPE, the Message Broker is a Cloud-Based Middleware (CBM) and Edge Coordinators (ECs) component.

VI. EVALUATION

The interaction model described in section IV is simulated and compared to four smart parking system baselines. The assessment metrics are driving time, distance, fuel consumption, CO2 emissions and parking search costs in Canadian dollars.

A. EXPERIENTIAL SETUP

In this assessment, we simulate SCOPE parking agents with a form of smart objects that implement the CIR-Agent model [34]. Each parking agent (PFA) that represents a parking facility has knowledge of available space in the facility and provides this knowledge to the Edge Coordinator (EC) upon request. Afterwards, the EC selects the best parking spot based on the driver's preferences.

To further elaborate on the simulation architecture, the SCOPE environment was simulated as a web application composed of a number of micro-services. This application is structured into three primary components: a Cloud-Based Middleware (CBM) service, a set of Edge Coordinator (EC) services, and multiple Parking Facility services (Smart Objects). In addition, Kafka, Eurka and Zull services are used to facilitate communication among SCOPE's primary components. Figure 7 graphically depicts SCOPE's simulation architecture. The simulation process is initiated when the CBM selects a random trip from the designated trips dataset. Subsequently, the CBM determines the appropriate Edge Coordinator responsible for facilitating interactions among the parking facilities within the geographical cluster corresponding to the trip's destination. Upon identification, the EC object communicates with potential parking facilities and orchestrates an auction based on the bids received. Once the auction concludes, the winning bid is relayed to the CBM object. This CBM then executes the baseline algorithms and saves the results in a MongoDB database for further analysis. It's important to highlight that SCOPE's geographical clustering serves as a method to arrange parking facilities into groups, where an edge coordinator manages the communication within the group. This arrangement doesn't interfere with the existing road network. Road distance and traffic details between the trip's start and nearby parking at the destination are sourced from the Google Distance Matrix API. This API optimally selects routes based on both historical and real-time data.

The experiments utilized 52 parking facilities in downtown Toronto, with coordinates collected from Google Places API [41]. For each experiment, 1050 trips are simulated in which drivers head to downtown Toronto where they search for parking. We use 105 trip origins, reflecting coordinates picked randomly in the Toronto suburbs (35 in East York, 35 in North York, and 35 in Etobicoke) and 105 random trip destinations in downtown Toronto. We employ the Google Distance Matrix API to calculate the time and distance a car travels while searching for a parking spot.

For each trip, we pass the following parameters to SCOPE and the baselines: a random trip origin, a random

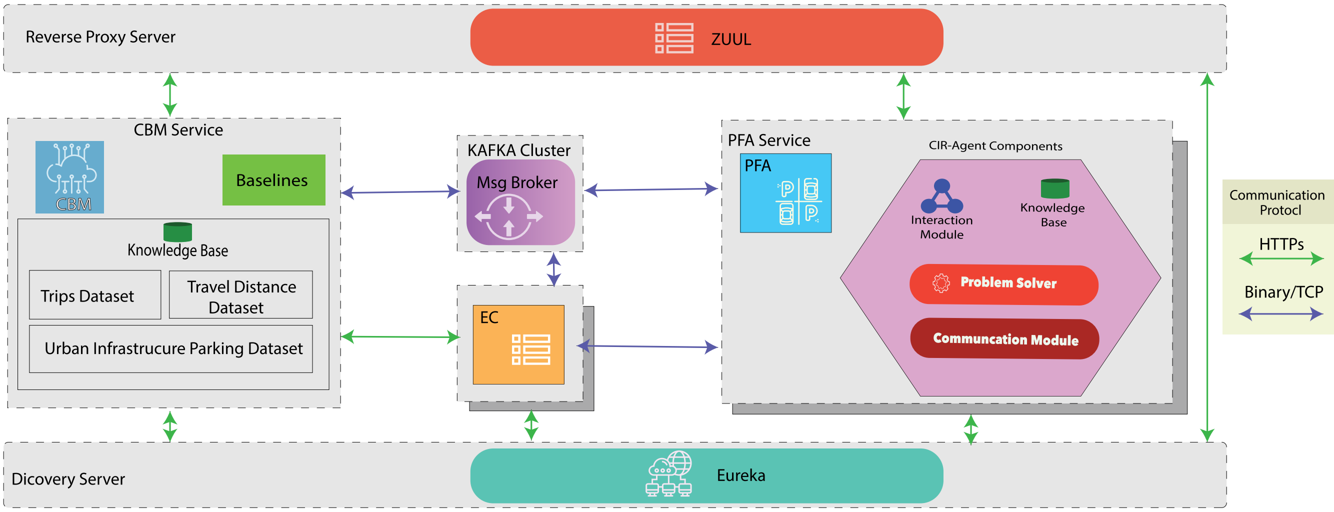

FIGURE 7. SCOPE simulation architecture.

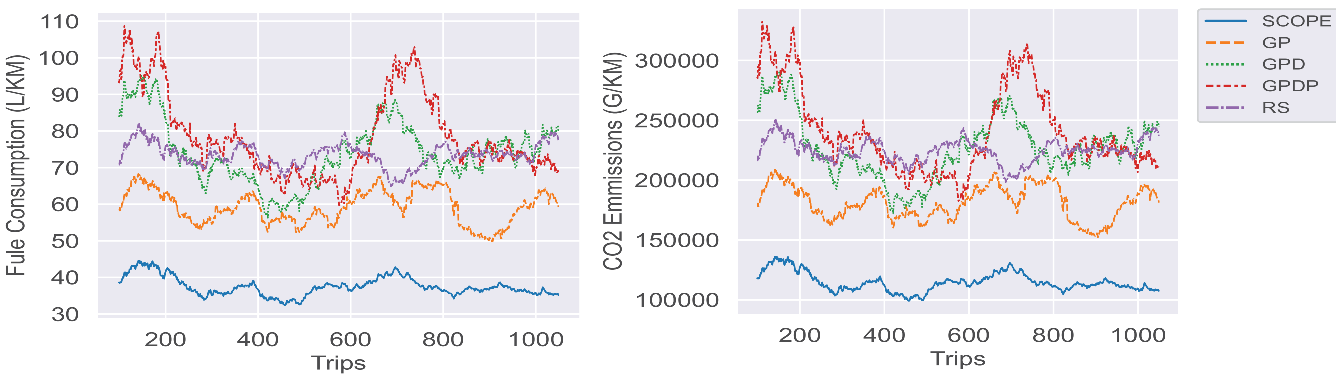

FIGURE 8. SCOPE vs baselines: environmental metrics (rolling avg window = 100 trip).

destination, a set of user preferences including (Walking Distance, Price, and Driving Distance), and a search radius of 1 kilometre. The performance of SCOPE and the baselines are measured in two settings based on user preferences. The values associated with walking distance are the following:

$$W = \{W_w : 0.6666667, W_p : 0.0666667, W_d : 0.266666667\}$$

and the values associated with the driving distance are the following:

$$W = \{W_w : 0.2666667, W_p : 0.0666667, W_d : 0.666666667\}$$

where W_w is the walking distance weight, W_p is the Price Weight, and W_d is the driving Distance Weight.

Fuel consumption and $CO2$ emissions are computed using International Energy Agency's report [42]. The report shows that Canadians consume an average of 8.9 litres of gasoline per 100 KM (8.9 L/100km), and cars in Canada produce an average of 206.0 g CO_2 per KM(206.0 g CO_2/KM). As for the fuel cost, we collected the average gasoline price in Toronto in 2021 [43].

B. URBAN INFRASTRUCTURE SEARCH FOR PARKING DATASET:

We use the search for parking dataset provided by GeoTab Inc. to supply baselines with traffic information [3]. The dataset highlights challenging parking areas in Toronto and is updated monthly, aggregating data over six months and summarized at Geohash level 7 (153m × 153m). We analyzed the dataset and constructed new features at Geohash levels 5, 6, and 7. The constructed features are an hourly distribution representing the number of vehicles searching for parking spots near a parking facility and an hourly distribution representing the total number of vehicles located around a parking lot. Figure 4 highlights the Geohash levels.

C. BASELINES

The baselines for SCOPE evaluation are probability-based parking systems inspired by the work proposed by Wu et al. [44], which features smart parking systems that have complete information about available parking spots in each facility. These systems utilize a probability model to determine the most appropriate parking facility, which is

FIGURE 9. SCOPE vs baselines: driving metrics (rolling avg window = 100 Trip).

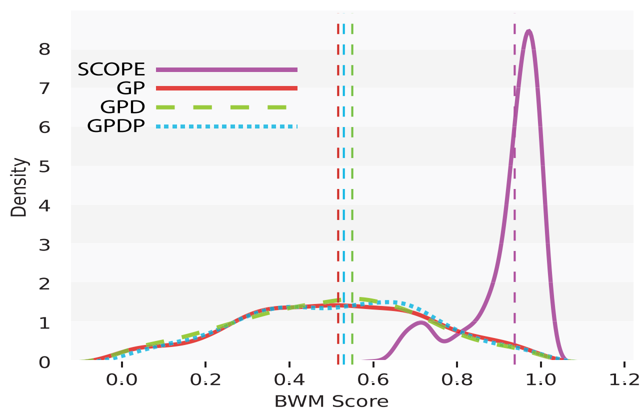

FIGURE 10. SCOPE vs baselines: driver satisfaction.

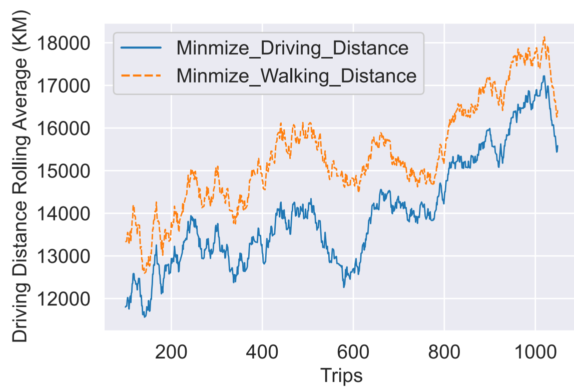

FIGURE 11. SCOPE behaviour reflecting driver preferences (rolling AVG window = 100 trip).

expected to have an available spot when the vehicle arrives. In this evaluation, the probability that there is available space at a given parking facility, i at time t, is computed as follows:

$$P_{as}(i,t) = 1 - \frac{occupancy(i,t)}{capacity(i,t)} \tag{35}$$

Furthermore, to add a level of smartness to the baseline, we use the Historical Urban Infrastructure dataset [3] to compute the competition for parking probability as follows:

$$P_{lc}(i,t) = 1 - \frac{\sigma(N_C(i,t)}{N_{AS}(i,t))} \tag{36}$$

where σ is the sigmoid function, $P_{lc}(i,t)$ is the probability of low competition for parking at a given parking facility i at time t, N_C represents the number of cars around the parking facility searching for spots, and N_{AS} is the number available spots in the parking facility, i, at time, t. Therefore, the probability that there is available space in a given parking facility at a given time can be formulated as follows:

$$P_\pi(i,t) = P_{as}(i,t) * P_{lc}(i,t) \tag{37}$$

1) RECOMMENDATION SYSTEM (RS)

Wu et al. [44] proposed a parking recommendation system that aims to minimize the expected travel time to successfully park a car. Given the driver's current location, RS arranges parking lots in a permutation set $\pi = \{PL_1, PL_2, \ldots, PL_n\}$ in which parking facilities around the car are arranged based on the probability of parking successfully in each parking facility. Assuming the driver follows the permutation set in order, the expected travel time is computed as follows:

For every parking facility PL_i, the time needed for a driver to reach the parking facility is weighted by the probability of successfully parking the car at the facility $PR(\pi, i)$. The same method is used for computing the expected travel distance. As this system provides an estimated driving time and distance for the permutation set, its comparison would SCOPE might not be fair. For this reason, the following greedy algorithms were implemented based on the permutation set computed by RS.

2) GREEDY BASED ON PROBABILITY (GP)

The aim of this algorithm is to simulate a car following the permutation set computed by RS until it successfully finds a parking spot. Whenever the car reaches a parking facility, a decision variable is assigned a value from the uniform distribution on the interval (0,1). If the decision variable is less than or equal to the probability that there is available space in PL_i (P_π(i)), then the system assumes the car finds

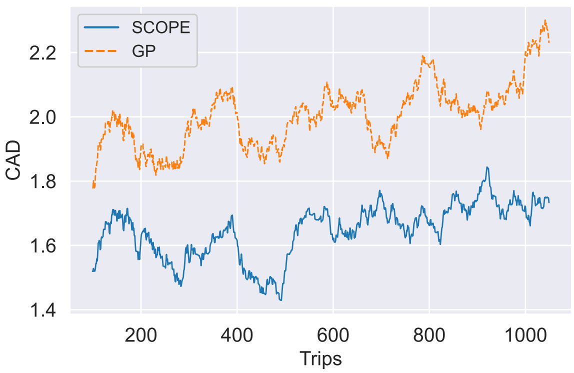

FIGURE 12. SCOPE vs GT: Trip cost in cad (rolling AVG window = 100 trip).

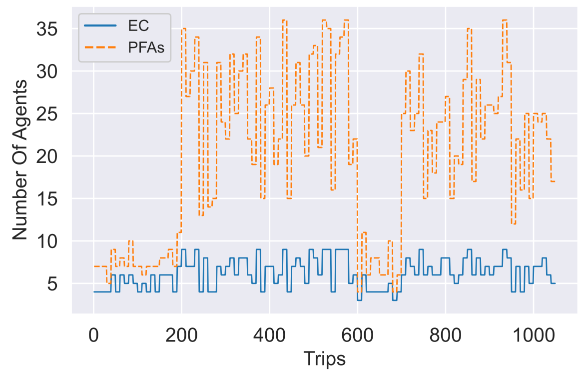

FIGURE 13. SCOPE agents interaction in response to requests.

a parking spot in PL_i. Otherwise, the car proceeds to the next parking lot in the permutation set. As the car travels from one parking facility to another, the system computes the travel time and distance using real data from Google Maps API. These values are added to the total travel time and total distance, respectively.

3) GREEDY BASED ON PROBABILITY AND DISTANCE (GPD)

This algorithm aims to minimize the travel distance. Generally, the GPD approach is similar to GP in the sense that there is a car following the permutation set until it successfully finds a parking spot. However, the permutation set is reordered for each decision cycle based on the following cost function.

$$Cost(PL_j) = P_\pi(j) distance(PL_i, PL_j) \tag{38}$$

4) GREEDY BASED ON PROBABILITY, DISTANCE, AND DRIVER PREFERENCES (GPDP)

The approach is similar to GPD. However, on every decision cycle, the permutation set is reordered based on an objective function that aims to minimize driving distance and takes into account the driver preferences.

$$utility(PL_j) = P_\pi(j) + BWM(preferences, PL_j) \tag{39}$$

$$Cost(PL_j) = utility(PL_j) * distance(PL_i, PL_j) \tag{40}$$

D. RESULTS

This section presents the performance of the baselines and the proposed approach used in SCOPE. The metrics used were environmental, driving satisfaction rate and the cost of finding a parking spot. Figure 9 shows the moving average of driving time and distance for each group of 100 trips, a 100-trip sliding window. The figure shows that SCOPE's average driving time was in the range of 35 minutes to 45 minutes, while SCOPE's driving distance was in the range of 12KM to 17 KM. The figure indicates that SCOPE outperformed the baselines in decreasing driving time and distance. Also, in terms of environmental metrics, SCOPE uses less fuel and emits less carbon monoxide, as shown in figure 8. The figure also shows that among the baselines, the GP model gives better results, but the difference between the GP and SCOPE results is large. Figure 12 shows that, on average, SCOPE trips cost 18 percent less than GP trips, with a mean moving average of 1.645 CAD for SCOPE and 2.004 CAD for GP.

$$PT(C, \pi) = \sum_{i=1}^{n} PR(\pi, i)(T_\pi(i) - t_0) \tag{41}$$

In addition, in comparison with the baselines, SCOPE can capture driver preferences and select parking spots that provide higher driver satisfaction rates. In this context, the driver satisfaction rate is measured using the parking facility BWM score. The Kernel Density Estimate (KDE) plot is used to visualize the distribution of BWM scores recorded by SCOPE and the baselines. Figure 10 depicts the KDE curve for SCOPE and the baselines. While the KDE curves for the baselines are comparable, the KDE curve for SCOPE demonstrates that SCOPE observations for BWM scores are significantly higher than those for the baselines.

Furthermore, figure 11 illustrates how SCOPE selection reflects the preferences of the driver. Figure 11 shows the moving average of SCOPE's driving distance in two separate experiments where the driver preferences for the exact trip origin and destination pair are weighted differently.

Finally, Figure 13 depicts SCOPE agents' cooperation to resolve the capability interdependency among ECs and PFAs; the fluctuation in the number of agents in every trip is a result of the variation of the number of PFAs around the trip's final destination.

Overall, the results show that developing a cooperative model that enables knowledge sharing and interaction among parking facilities can significantly reduce the travel distance and time a car spends searching for a parking spot.

VII. DISCUSSION ON LIMITATIONS AND REAL-WORLD DEPLOYMENT

While we view SCOPE system as a meaningful contribution to the field of smart parking management, it is essential to recognize that there are limitations and areas for further exploration to validate the approach comprehensively.

A. RELIABILITY AND FAULT TOLERANCE

The distributed nature of SCOPE, involving multiple interconnected components like Cloud-Based Middleware (CBM), Edge Coordinators (ECs), and Parking Facility Agents (PFAs), necessitates a robust framework for ensuring system reliability and fault tolerance. In a real-world scenario, the failure or latency in any of these components could have cascading effects, impacting the system's overall performance and user experience. This is particularly critical given SCOPE's real-time decision-making requirements for parking spot allocation. While the current work focuses on the interaction model, it does not delve into deployment-related challenges such as reliability and fault tolerance. Future work could address these limitations by exploring redundancy measures, failover mechanisms, and real-time monitoring to enhance the system's resilience and reliability.

B. CONSIDERATIONS ON TRUST AND PRIVACY PRESERVATION

Trust evaluation and privacy preservation are crucial yet challenging elements in facilitating data sharing in cooperative environments. These factors often have conflicting requirements, making it essential to find an optimal balance [45]. The computational, communication, and storage needs of any proposed solution can also influence its practicality. Various management schemes have been put forth to address these challenges. Notably, Liu et al. [45] introduced the Lightweight Privacy-Preserving Trust Evaluation (LPPTE) scheme, designed to balance trust and privacy while minimizing system overhead effectively. Given that SCOPE operates as a cooperative system where resources are traded and shared among different smart agents, incorporating trust and privacy management schemes, such as LPPTE [45], BTMPP [46], TROVE [47] and PPRM [48] could significantly enhance the system's value. Although these aspects are beyond the scope of the current study, future research could further consider integrating cryptographic schemes, reputation models, and privacy-preserving algorithms to enhance the system's trustworthiness and data security.

C. REAL-WORLD DEPLOYMENT

The following discussion focuses on key considerations necessary for evolving SCOPE from a theoretical construct into a practical solution suitable for deployment in smart city settings.

1) SMART CITY INTEGRATION

SCOPE fits well within the smart city paradigm, where IoT devices, sensors, and intelligent algorithms work in tandem to optimize urban services. Integration with existing smart city platforms could be a viable path for deployment.

2) 5G NETWORKS AND TELECO CLOUD

The advent of 5G networks offers unprecedented low-latency and high-bandwidth capabilities, which are crucial for real-time decision-making in intelligent systems like SCOPE. However, to fully exploit these advantages, it is essential to bring computation closer to the data source, thereby reducing the time data travel back and forth between the cloud and the edge of the network. This is where AWS Wavelength comes into play. AWS Wavelength extends AWS infrastructure to telecom networks, enabling developers to run latency-sensitive applications closer to end-users [49]. By hosting SCOPE's Edge Coordinators (ECs) on Wavelength-enabled 5G networks, several key benefits can be realized, such as Reduced Latency, Localized Data Traffic, Real-time Analytics, Scalability and Flexibility, and Enhanced Reliability. The synergy between 5G networks and AWS Wavelength can significantly enhance SCOPE's performance, making it a highly responsive and reliable system for smart parking management. This integration aligns well with the broader vision of smart cities, where low-latency, high-reliability services are essential for improving urban living conditions.

3) INTEGRATION WITH AUTONOMOUS VEHICLES

As autonomous vehicles become more prevalent, SCOPE could serve as the parking management component within a broader autonomous vehicle ecosystem. SCOPE can benefit from V2X communication technologies, enabling more dynamic interactions between vehicles and parking facilities.

VIII. CONCLUSION

This work presents SCOPE, a multi-agent smart distributed cooperative system that addresses parking challenges in a smart city environment by fostering coordination, autonomous interaction, and resource sharing. SCOPE leverages an overlay network, a hierarchical and spatial structure of coordination nodes, and an integration layer to enable seamless communication and cooperation among parking agents. The proposed architecture facilitates a sharing economy business model, allowing private parking owners to monetize their unused spaces, maximizing resource usage, and providing affordable access to parking. By considering driver preferences and enabling autonomous interaction among parking systems, SCOPE effectively reduces search time, traffic congestion, and environmental impact. This work fills a gap in the literature by providing a comprehensive solution to the parking challenges in smart city settings, emphasizing agent cooperation and interaction in an open environment. For future research, there are pivotal areas to address. Firstly, the system's distributed nature necessitates a deeper focus on reliability and fault tolerance, possibly through redundancy measures and real-time monitoring. Secondly, the cooperative essence of SCOPE calls for a balance between trust and privacy, which might be achieved by incorporating cryptographic schemes and reputation models. Lastly, for real-world deployment, integrating SCOPE with smart city platforms, harnessing 5G networks and edge computing services like AWS Wavelength, and exploring SCOPE's role within the autonomous vehicle ecosystem using V2X communication technologies will be crucial. Ultimately, SCOPE holds the potential to revolutionize the way parking is managed in smart cities, improving the quality of life for urban residents and contributing to a more sustainable future.

REFERENCES

[1] INRIX. (2017). *Searching for Parking Costs the UK $23.3 Billion a Year*. Accessed: Mar. 15, 2022. [Online]. Available: http://inrix.com/pressreleases/parking-pain-uk/

[2] S. Mathur, T. Jin, N. Kasturirangan, J. Chandrasekaran, W. Xue, M. Gruteser, and W. Trappe, "ParkNet: Drive-by sensing of road-side parking statistics," in *Proc. 8th Int. Conf. Mobile Syst., Appl., Services*, Jun. 2010, pp. 123–136.

[3] GeoTab. (2008). *Urban Infrastructure: Search for Parking Dataset*. Accessed: Mar. 15, 2022. [Online]. Available: https://data.geotab.com/urban-infrastructure/searching-for-parking

[4] F. P. Appio, M. Lima, and S. Paroutis, "Understanding smart cities: Innovation ecosystems, technological advancements, and societal challenges," *Technolog. Forecasting Social Change*, vol. 142, pp. 1–14, May 2019.

[5] R. Khatoun and S. Zeadally, "Smart cities: Concepts, architectures, research opportunities," *Commun. ACM*, vol. 59, no. 8, pp. 46–57, Jul. 2016.

[6] D. Ahlers, P. Driscoll, E. Löfström, J. Krogstie, and A. Wyckmans, "Understanding smart cities as social machines," in *Proc. 25th Int. Conf. Companion World Wide Web WWW Companion*, 2016, pp. 759–764.

[7] Y. D. Wang, H. Ghenniwa, and W. Shen, "Ontological view based semantic transformation for distributed systems," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, Oct. 2009, pp. 4007–4012.

[8] S. R. Rizvi, S. Zehra, and S. Olariu, "ASPIRE: An agent-oriented smart parking recommendation system for smart cities," *IEEE Intell. Transp. Syst. Mag.*, vol. 11, no. 4, pp. 48–61, Winter 2019.

[9] S. R. Rizvi, S. Zehra, and S. Olariu, "MAPark: A multi-agent auction-based parking system in Internet of Things," *IEEE Intell. Transp. Syst. Mag.*, vol. 13, no. 4, pp. 104–115, Winter 2021.

[10] J. Babic, A. Carvalho, W. Ketter, and V. Podobnik, "Evaluating policies for parking lots handling electric vehicles," *IEEE Access*, vol. 6, pp. 944–961, 2018.

[11] L. Rahmani, D. Minarsch, and J. Ward, "Peer-to-peer autonomous agent communication network," in *Proc. 20th Int. Conf. Auto. Agents MultiAgent Syst.*, 2021, pp. 1037–1045.

[12] M. Ritter and H. Schanz, "The sharing economy: A comprehensive business model framework," *J. Cleaner Prod.*, vol. 213, pp. 320–331, Mar. 2019.

[13] W. Griggs, J. Y. Yu, F. Wirth, F. Häusler, and R. Shorten, "On the design of campus parking systems with QoS guarantees," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 5, pp. 1428–1437, May 2016.

[14] A. Hira and K. Reilly, "The emergence of the sharing economy: Implications for development," *J. Developing Societies*, vol. 33, no. 2, pp. 175–190, Jun. 2017.

[15] M. Khalid, K. Wang, N. Aslam, Y. Cao, N. Ahmad, and M. K. Khan, "From smart parking towards autonomous valet parking: A survey, challenges and future works," *J. Netw. Comput. Appl.*, vol. 175, Feb. 2021, Art. no. 102935.

[16] M. Aljohani, S. Olariu, A. Alali, and S. Jain, "A survey of parking solutions for smart cities," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 10012–10029, Aug. 2022.

[17] P. Melnyk, S. Djahel, and F. Nait-Abdesselam, "Towards a smart parking management system for smart cities," in *Proc. IEEE Int. Smart Cities Conf. (ISC)*, Oct. 2019, pp. 542–546.

[18] G. N. Hainalkar and M. S. Vanjale, "Smart parking system with pre & post reservation, billing and traffic app," in *Proc. Int. Conf. Intell. Comput. Control Syst. (ICICCS)*, Jun. 2017, pp. 500–505.

[19] T. N. Pham, M.-F. Tsai, D. B. Nguyen, C.-R. Dow, and D.-J. Deng, "A cloud-based smart-parking system based on Internet-of-Things technologies," *IEEE Access*, vol. 3, pp. 1581–1591, 2015.

[20] G. Baranwal, D. Kumar, and D. P. Vidyarthi, "A multi-criteria framework for smart parking recommender system," in *Proc. IEEE Int. Smart Cities Conf. (ISC)*, Sep. 2020, pp. 1–8.

[21] D. Kanteti, D. V. S. Srikar, and T. K. Ramesh, "Intelligent smart parking algorithm," in *Proc. Int. Conf. Smart Technol. Smart Nation (SmartTechCon)*, Aug. 2017, pp. 1018–1022.

[22] T. Delot, N. Cenerario, S. Ilarri, and S. Lecomte, "A cooperative reservation protocol for parking spaces in vehicular ad hoc networks," in *Proc. 6th Int. Conf. Mobile Technol., Appl. Syst. Mobility*, 2009, pp. 1–8.

[23] A. O. Kotb, Y.-C. Shen, X. Zhu, and Y. Huang, "IParker—A new smart car-parking system based on dynamic resource allocation and pricing," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 9, pp. 2637–2647, Sep. 2016.

[24] J. Lin, S.-Y. Chen, C.-Y. Chang, and G. Chen, "SPA: Smart parking algorithm based on driver behavior and parking traffic predictions," *IEEE Access*, vol. 7, pp. 34275–34288, 2019.

[25] P. Zhao, H. Guan, and P. Wang, "Data-driven robust optimal allocation of shared parking spaces strategy considering uncertainty of public users' and owners' arrival and departure: An agent-based approach," *IEEE Access*, vol. 8, pp. 24182–24195, 2020.

[26] H. Dargaye, B. Gobin-Rahimbux, and N. G. Sahib-Kaudeer, "Agent-based modelling for a smart parking system for mauritius," in *Information Systems Design and Intelligent Applications*. Cham, Switzerland: Springer, 2019, pp. 367–377.

[27] L. Wang, H. Chen, and Y. Li, "An active parking guidance information system based on dynamic agent negotiation," in *Proc. ICCTP*, Jul. 2009, pp. 1–6.

[28] I. Benenson, K. Martens, and S. Birfir, "PARKAGENT: An agent-based model of parking in the city," *Comput., Environ. Urban Syst.*, vol. 32, no. 6, pp. 431–439, Nov. 2008.

[29] M.-Y. Hsieh, Y. Lai, H. Y. Lin, and K.-C. Li, "A model for predicting vehicle parking in fog networks," in *Proc. Int. Conf. Frontier Comput.* Cham, Switzerland: Springer, 2016, pp. 239–249.

[30] C. Tang, X. Wei, C. Zhu, W. Chen, and J. J. P. C. Rodrigues, "Towards smart parking based on fog computing," *IEEE Access*, vol. 6, pp. 70172–70185, 2018.

[31] R. Tandon and P. Gupta, "Optimizing smart parking system by using fog computing," in *Proc. Int. Conf. Adv. Comput. Data Sci.* Cham, Switzerland: Springer, 2019, pp. 724–737.

[32] K. S. Awaisi, A. Abbas, M. Zareei, H. A. Khattak, M. U. S. Khan, M. Ali, I. U. Din, and S. Shah, "Towards a fog enabled efficient car parking architecture," *IEEE Access*, vol. 7, pp. 159100–159111, 2019.

[33] A. Dafoe, Y. Bachrach, G. Hadfield, E. Horvitz, K. Larson, and T. Graepel, "Cooperative AI: Machines must learn to find common ground," *Nature*, vol. 593, no. 7857, pp. 33–36, 2021.

[34] H. Ghenniwa and M. Kamel, "Interaction devices for coordinating cooperative distributed systems," *Intell. Automat. Soft Comput.*, vol. 6, no. 3, pp. 173–184, 2000.

[35] M. Gansterer, R. F. Hartl, and M. Savelsbergh, "The value of information in auction-based carrier collaborations," *Int. J. Prod. Econ.*, vol. 221, Mar. 2020, Art. no. 107485.

[36] D. C. Wyld, "Current research on reverse auctions: Part I—Understanding the nature of reverse auctions and the price and process savings associated with competitive bidding," *Int. J. Manag. Value Supply Chains*, vol. 2, no. 3, pp. 11–23, Sep. 2011.

[37] R. Tassabehji, "Understanding e-auction use by procurement professionals: Motivation, attitudes and perceptions," *Supply Chain Manag., Int. J.*, vol. 15, no. 6, pp. 425–437, Sep. 2010.

[38] G. Niemeyer. (2008). *Geohash: Geohash is the Encoding Technique Used to Convert Geoareas Into Base-32 Alphabet Encoding*. Accessed: Mar. 15, 2022. [Online]. Available: https://geohash.org

[39] R. Moussalli, M. Srivatsa, and S. Asaad, "Fast and flexible conversion of geohash codes to and from latitude/longitude coordinates," in *Proc. IEEE 23rd Annu. Int. Symp. Field-Program. Custom Comput. Mach.*, May 2015, pp. 179–186.

[40] J. Rezaei, "Best-worst multi-criteria decision-making method: Some properties and a linear model," *Omega*, vol. 64, pp. 126–130, Oct. 2016.

[41] Google. (2021). *Places API*. Accessed: Mar. 15, 2022. [Online at https://developers.google.com/maps/documentation/places/web-service/overview

[42] C. E. Regulator. (2017). *Market Snapshot: How Does Canada Rank in Terms of Vehicle Fuel Economy*. Accessed: Mar. 15, 2022. [Online]. Available: https://www.cer-rec.gc.ca/en/data-analysis/energy-markets/market-snapshots/2019/market-snapshot-how-does-canada-rank-in-terms-vehicle-fuel-economy.html

[43] S. Canada. (2021). *Monthly Average Retail Prices for Gasoline and Fuel Oil, by Geography*. Accessed: Mar. 15, 2022. [Online]. Available: https://www150.statcan.gc.ca/t1/tbl1/en/tv.action?pid=1810000101

[44] E. H. Wu, J. Sahoo, C.-Y. Liu, M.-H. Jin, and S.-H. Lin, "Agile urban parking recommendation service for intelligent vehicular guiding system," *IEEE Intell. Transp. Syst. Mag.*, vol. 6, no. 1, pp. 35–49, Spring 2014.

[45] Z. Liu, J. Ma, J. Weng, F. Huang, Y. Wu, L. Wei, and Y. Li, "LPPTE: A lightweight privacy-preserving trust evaluation scheme for facilitating distributed data fusion in cooperative vehicular safety applications," *Inf. Fusion*, vol. 73, pp. 144–156, Sep. 2021.

[46] Z. Liu, F. Huang, J. Weng, K. Cao, Y. Miao, J. Guo, and Y. Wu, "BTMPP: Balancing trust management and privacy preservation for emergency message dissemination in vehicular networks," *IEEE Internet Things J.*, vol. 8, no. 7, pp. 5386–5407, Apr. 2021.

[47] J. Guo, X. Li, Z. Liu, J. Ma, C. Yang, J. Zhang, and D. Wu, "TROVE: A context-awareness trust model for VANETs using reinforcement learning," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6647–6662, Jul. 2020.

[48] Y. Cheng, J. Ma, Z. Liu, Y. Wu, K. Wei, and C. Dong, "A lightweight privacy preservation scheme with efficient reputation management for mobile crowdsensing in vehicular networks," *IEEE Trans. Dependable Secure Comput.*, vol. 20, no. 3, pp. 1771–1788, 2023, doi: 10.1109/TDSC.2022.3163752.

[49] I. Amazon Web Services. (2023). *AWS Wavelength*. Accessed: Aug. 30, 2023. [Online]. Available: https://aws.amazon.com/wavelength/

MUHAMED ALARBI (Member, IEEE) received the B.Sc. degree in computer science from the University of Tripoli, Libya, in 2011, and the M.Sc. degree in computer science from The University of Western Ontario (UWO), London, ON, Canada, in 2017, where he is currently pursuing the Ph.D. degree with the Department of Computer Science. He was passionate about middleware architectures for smart open environments. His research interests include edge computing, service management and orchestration, resource allocation, and the development of edge computing frameworks for smart cities and cooperative AI systems.

ABDELKAREEM JARADAT (Member, IEEE) received the B.Sc. degree (Hons.) in computer engineering from the Jordan University of Science and Technology (JUST), Irbid, Jordan, in 2009, and the M.Sc. degree in computer science from The University of Western Ontario (UWO), London, ON, Canada, in August 2019, where he is currently pursuing the Ph.D. degree with the Department of Computer Science. His primary research interests include time-series analysis, machine learning, the Internet of Things, smart home energy management systems, and the application of information visualization focusing on smart services and applications, including demand response, smart city/home, and smart grids.

HANAN LUTFIYYA (Senior Member, IEEE) is currently a Professor with the Department of Computer Science, The University of Western Ontario (UWO), London, ON, Canada. Her research interests include the Internet of Things, software engineering, self-adaptive and self-managing systems, autonomic computing, monitoring and diagnostics, mobile systems, policies, and clouds. She was a recipient of the UWO Faculty Scholar Award, in 2006. She is currently the Editor-in-Chief of the IEEE Transactions on Network and Service Management. She has served as the General Co-Chair for the IEEE International Conference on Network and Service Management. She is currently on the board of directors for CS-Can—Info-Can. She is a Past Member of the Natural Science and Engineering Research Council of Canada (NSERC) Discovery Grant Committee and a Past Member and the Chair of an NSERC Strategic Grants Committee. She was a member of the Computer Science Accreditation Council.

ANWAR HAQUE (Member, IEEE) is currently an Associate Professor and the Undergraduate Chair of the Department of Computer Science, The University of Western Ontario (UWO). Before joining UWO, he was the Associate Director of Bell Canada. He has authored/coauthored more than 100 peer-reviewed research publications in leading journals and conferences, written many industry technical papers, holds several patents/licenses, and supervised more than 100 graduate students/highly qualified professionals. His primary research interests include next-generation communication networks (such as 5G and beyond), the IoT, and cyber-security, focusing on autonomous systems, and smart services and applications. He has been awarded several national/provincial-level research grants, including NSERC, MITACS, OCE, and SOSCIP. His collaborative research grants are valued at more than $15 million. At UWO, he served as the industry expert-in-residence with the Faculty of Science, a member of the Western Senate, and on the inaugural advisory committee for the newly established Bell-Western 5G Research Centre. He is the Director of the Western Information and Networking Group (WING) Laboratory, where he conducts cutting-edge research in emerging network technologies and smart systems.