IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 2 December 2023, accepted 21 December 2023, date of publication 3 January 2024,
date of current version 10 January 2024.

Digital Object Identifier 10.1109/ACCESS.2023.3349376

SURVEY

Smart City Middleware: A Survey and a Conceptual Framework

CHRISTOS GOUMOPOULOS, (Member, IEEE)
Department of Information and Communications Systems Engineering, University of the Aegean, 832 00 Samos, Greece
e-mail: goumop@aegean.gr

This work was partly supported by the University of the Aegean Research Unit through the "Internet of Things-Intelligent Environments in Next-Generation Networks" Project under Grant Id 70477.

ABSTRACT Smart city middleware serves as a foundational tool in the evolution of urban digitalization, acting as an intermediary software layer that simplifies the development, deployment, and management of applications tailored for smart urban environments. However, the development of effective middleware for smart cities is challenging. The present research embarks on a comprehensive exploration of the smart city middleware landscape, unraveling the intricacies of its development and the challenges faced therein. Rooted in the assessment of 20 distinct middleware solutions, our study highlights the pivotal technologies, features and functionalities that are imperative for a middleware to effectively support a city's digital transformation. The functional and non-functional requirements form the nucleus of our evaluation. We also explore the architectural styles pivotal to middleware development and the programming paradigms shaping smart city application development. Our study highlights challenges in using middleware for smart city applications, such as interoperability, scalability, security amidst big data, context management, reliability, quality of service, energy efficiency, and compliance with technological standards and regulations. Based on the detailed analysis, we propose a conceptual framework for smart city middleware, shaped by the challenges and requirements identified in existing literature and middleware solutions. This framework is designed to reflect the diverse demands and complexities of urban digital transformation, and guide smart city middleware development accordingly. As a result, this research stands as a reference study for software developers, urban planners, and researchers, outlining the current state and future directions in the domain of smart city middleware.

INDEX TERMS Smart cities, middleware, conceptual framework, surveys, functional and non-functional requirements, enabling technologies, architectural styles, programming paradigms, challenges.

I. INTRODUCTION

Smart city middleware, a key enabler in the digital transformation of urban landscapes, streamlines the creation and management of smart city applications, though its development presents significant challenges, as explored in this research study.

This section presents the driving forces behind the adoption of smart city middleware, highlighting the research questions explored in this survey. It then provides an overview of existing literature, offering context and contrasting this study with previous research, thereby outlining the unique contribution of this work. Finally, it presents the structure of the paper, guiding readers through a methodical exploration of smart city middleware solutions.

The associate editor coordinating the review of this manuscript and approving it for publication was Mohammad Ayoub Khan.

A. MOTIVATION

A smart city is essentially the consolidation of numerous smart solutions that span across all sectors of society, working together to improve the quality of life, increase operational efficiency, and foster sustainable development [1], [2]. Various smart systems have been put forward for numerous urban services, including public utilities [3], [4], transportation [5], healthcare [6], environment [7], public safety [8], education [9] and governance [10].

The realization of such diverse systems is fundamentally facilitated by a comprehensive suite of enabling technologies. Information and Communication Technologies (ICT) serve as the underlying infrastructure for these smart solutions, empowering real-time data collection, communication, and analysis, thereby enhancing various urban services and supporting the broader vision of smart cities [11]. Internet of Things (IoT) devices, including sensors and actuators, are critical for monitoring urban environments and infrastructure, and for executing automated responses [12]. Advanced data analytics, including Artificial Intelligence (AI) and machine learning, turn the large volumes of data generated by these devices into actionable insights [13]. For instance, these insights can be used to optimize traffic flow, manage energy consumption, and predict crime. Moreover, technologies like cloud computing and edge computing provide the necessary infrastructure for data storage and processing [14], while Blockchain technology can enhance security and transparency, particularly in digital governance [10].

While many existing solutions are specialized, focusing on a specific domain or addressing a singular problem, and often developed from scratch with minimal software reuse, this approach does not sufficiently cater to the comprehensive requirements of a smart city environment [15]. The creation of independent, vertical applications for each city domain might address individual urban challenges, but this approach fails to tap into the potential synergies across domains [16]. A smart city thrives on integrated solutions that can interact and collaborate, thereby maximizing efficiency and optimizing resource utilization across the city's various sectors. Therefore, while domain-specific applications have their merits, it is the horizontal integration of these applications through common software platforms that truly drives the realization of a smart city [17].

Smart city middleware is an integrated software platform that acts as a supportive environment for software developers [18]. It assists in the process of designing, implementing, deploying, and managing applications specifically tailored for smart cities. It acts as an intermediate layer, facilitating communication, interoperability, and data integration between diverse systems and applications within a smart city's digital ecosystem. This middleware provides the necessary tools and interfaces for software developers to effectively and efficiently build and manage applications that enhance a city's intelligence and responsiveness.

The construction of smart city middleware poses significant challenges, among which ensuring interoperability stands as a prominent issue [19]. This interoperability refers to the seamless communication and cooperation between an array of diverse components, including different systems, applications, and devices, possibly developed by various vendors utilizing a multitude of platforms and protocols. Other key challenges include [15], [17], [20]: scalability, which requires the middleware to accommodate the city's expansion and increasing data volumes; security and privacy, involving stringent encryption protocols and compliance with data protection regulations due to the vast amounts of data involved; real-time processing to support public safety or traffic management applications; reliability and fault tolerance to prevent severe disruptions from system downtime; integration capabilities to seamlessly align with a range of applications; effective data management strategies to handle abundant data volumes from multiple sensors and sources; and lastly, strict adherence to diverse technological standards and regulatory guidelines across different regions.

In this research endeavor, a rigorous evaluation is conducted of efforts focused on the development of middleware solutions for smart cities, which includes a review of relevant requirements as emphasized in existing literature. From this analysis, a conceptual framework has been derived, adeptly mapping considerations related to these requirements. In order to deepen understanding and identify the characteristics that smart city middleware should embody to facilitate the creation of comprehensive smart city applications, an analysis of 20 middleware solutions, targeted towards smart city implementations, has been undertaken.

Guided by the objective to build comprehensive insights, the study formulated and explored four specific research questions, outlined as follows:

RQ1: Which enabling technologies are integral to the functionality of smart city middleware?

RQ2: What functional and non-functional requirements characterize smart city middleware?

RQ3: What are the principal architectural styles and programming paradigms integrated in smart city middleware?

RQ4: What challenges and open issues preside over the future of smart city middleware?

B. RELATED WORK

The literature contains a number of pertinent reviews or surveys that address the topics of software platforms and architectures for smart cities. Razzaque et al. provide a comprehensive overview of IoT middleware, covering key IoT characteristics, middleware requirements, a review of existing systems, and highlighting open research challenges in this domain [21]. In their comprehensive analysis, Ketu & Mishra highlight the pivotal role of IoT as a cornerstone in shaping smart city paradigms and the intricacies it introduces [22]. The paper delves into current trends, structural architectures, and challenges, offering valuable insights for both researchers and practitioners in the realm of smart urban development. Abadía et al. aim to elucidate the role of IoT frameworks in smart city applications through an extensive survey of contemporary IoT frameworks designed for urban contexts [23]. Their work also introduces an abstract IoT framework concept, which may influence future implementations in smart cities. Astropekakis et al. [24] conducted a survey of IoT software platforms tailored for developing scalable and efficient smart city applications. Their study introduces a comprehensive evaluation scheme for these platforms, categorizing them into core, data management, application empowerment, and accessibility criteria. Such studies primarily focus on the role

of IoT and related software platforms or frameworks in smart city development without delving into the broader areas this study emphasizes.

The work of Santana et al. examines the role of ICT in advancing smart city environments, highlighting the potential and challenges associated with smart city software platforms [17]. Analyzing several projects, the study delineates a reference architecture for future platform development, categorizes core enabling technologies, and addresses prevalent research challenges, providing guidance for various stakeholders in smart city initiatives. The survey by Farahzadi et al. explores the challenges posed by the heterogeneity of objects in Cloud of Things (CoT) platforms, emphasizing the pivotal role of middleware as a solution for seamless communication between diverse entities [25]. Through an exploration of middleware technologies, architectural styles, and service domains, the study presents suitable middleware for CoT while highlighting existing challenges and concerns in their design. The review performed by Alasbali et al. thoroughly examines intersections between Blockchain technologies, IoT functionalities, and smart city solutions, revealing that Blockchain's intermediary role offers significant enhancements over previous structural constraints, particularly in areas like security, authentication, and smart contract application [26]. Such surveys primarily focus on the technological intricacies of smart city platforms, software platforms design, and the potential synergy between established and emergent technologies like Blockchain and IoT in urban contexts.

Table 1 provides a comparative overview of related surveys detailing their scope, evaluation dimensions, challenges identified, and main outcomes.

C. CONTRIBUTION

Although we share some common ground with related research, particularly in the examination of underlying technologies and challenges, our approach diverges due to its comprehensive perspective, delving deeper into the breadth of the research questions explored in this study. In this research, we analyzed 20 contemporary smart city middleware solutions to identify key functional and non-functional requirements as well as the associated challenges in this field. Additionally, we provide insights regarding their architectural design and the programming paradigms adopted to facilitate application development. Based on the requirements highlighted by the examined studies, we proposed a conceptual framework. This framework outlines functionalities prevalent in existing solutions and highlights essential topics that should be considered for the optimal development of future smart city middleware. Overall, this study provides valuable insights that will be beneficial to the different stakeholders engaged in smart city development.

D. OUTLINE

Section II introduces the smart city middleware solutions examined in this survey. In addressing RQ1, a review of the literature was conducted to identify the key enabling technologies utilized in smart city middleware, as elaborated upon in Section III. In response to RQ2, a qualitative analysis of the smart city middleware solutions introduced in Section II was undertaken, with the primary functional and non-functional requirements detailed in Section IV. For RQ3, an in-depth examination of the design philosophies underpinning smart city middleware development and their programming models supporting application development was conducted, as discussed in Section V. RQ4 prompted the study to delve into existing research, identifying forthcoming challenges and outstanding issues in smart city middleware, setting the stage for future endeavors as illustrated in Section VI. The synthesis of findings from the research questions led to the development of a conceptual framework, elucidating key topics essential for constructing an optimal smart city middleware, as detailed in Section VII. A discussion is provided with insights and perspectives on the intricacies of smart city middleware in Section VIII. Conclusions are then encapsulated in Section IX.

II. MIDDLEWARE SOLUTIONS FOR SMART CITIES

In this section, various smart city middleware platforms from the literature are discussed. To identify appropriate studies, the subsequent query string was employed: (''Smart City'' OR ''Smart Cities'') AND (Middleware OR ''Software Platform'' OR Architecture). The review methodology incorporated articles from both journals and conference proceedings, drawing from prominent computer science digital repositories such as IEEE Digital Library, ACM Digital Library, Science Direct, Springer Link, Scopus, and Google Scholar. Using the specified search terms within the designated digital libraries, 278 articles were identified based on a review of their titles, abstracts and keywords. Upon further examination of the selected papers, this study concentrated on 41 articles that delineate smart city middleware solutions. The shortlisted papers for the final analysis were selected based on criteria including relevance to the specific search terms, answering at least two research questions, and being peer-reviewed; while papers not in English, published before 2010, of minimal length, or containing solutions unsuitable for smart city contexts were excluded.

A. AMF-CPS

AMF-CPS is an agent-based middleware framework (AMF) that leverages distributed Cyber-Physical Systems (CPS) to enhance communication reliability within smart city environments [27]. Agent-based technology in the distributed CPS, operating autonomously and independently, mitigates data source downtime challenges, ensuring consistent resource sharing and response scheduling irrespective of query demands and communication durations. This agent-based framework was designed to function as middleware, bridging the application and platform layers to enhance interoperability. The middleware incorporates various agents, including resource managers, access controllers, and schedulers. These agents, essentially defined as compact programs or rule sets,

TABLE 1. Comparative overview of related surveys.

Survey	Scope	Evaluation Dimensions	Challenges Identified	Main Outcomes
Razzaque et al. [21]	IoT middleware focusing on IoT characteristics and system requirements.	Functional and non-functional requirements; Architectural styles.	Functional: Resource discovery & management; Data management; Event management; Code management. Non-Functional: Real time; Reliability; Availability; Security & privacy; Ease-of-deployment; Popularity. Architectural: Programming abstraction; Interoperability; Adaptation; Context-awareness and autonomous behavior.	An evaluation of existing IoT middleware solutions, which are organized in 7 groups based on their design approach.
Ketu & Mishra [22]	IoT architectural frameworks in the context of smart cities.	Smart city domains; IoT technologies and protocols; Machine learning and deep learning approaches.	Privacy & security; Reliability; Heterogeneity; Scalability; Sensor networks; Big Data; Open Barriers; Legal & social aspects.	30 real-world examples of IoT involvement in smart cities.
Abadía et al. [23]	IoT frameworks used in smart city applications.	Smart city domains; IoT technologies and protocols; Functional and non-functional requirements.	Scalability; Interoperability.	A review of 30 IoT frameworks; An abstract concept for an IoT framework tailored for smart city applications.
Astropekakis et al. [24]	IoT software platforms for scalable smart city applications.	Functional and non-functional requirements.	Standardization; Energy management; Security & privacy; scalability; Management & self-configuration; Data storage & processing; Availability & reliability; Network performance & Quality of Service; Interoperability.	An evaluation scheme comprising 4 categories of metrics: Core, Data Management & Processing, Application Empowerment, Accessibility; A comparative analysis of 15 IoT software platforms.
Santana et al. [17]	State of the art in smart city software platforms.	Smart city domains; Enabling technologies; Functional and non-functional requirements.	Privacy; Data management; Heterogeneity; Energy management; Communication; Scalability; Security; Lack of testbeds; City models; Platform maintenance.	An analysis of 23 projects related to software platforms for smart cities; A reference architecture to guide the development of future smart city software platforms.
Farahzadi et al. [25]	Middleware technologies suited for Cloud of Things-based platforms.	Smart city domains; Enabling technologies; Functional and non-functional requirements; Architectural styles.	Real time; Resource discovery; Security & privacy; Lightweight footprint; Data analysis; Quality of Service; Standardization.	An evaluation of 20 middleware solutions.
Alasbali et al. [26]	Intersection of Blockchain technologies, IoT capabilities, and smart city solutions	Non-functional requirements for securing and protecting IoT-based integration into smart city solutions through Blockchain technology.	Security and authentication in IoT; Smart contract technologies for decentralized smart negotiation; Managing the heterogeneity and proprietary nature of technologies and their impact on IoT's scope and capabilities.	An evaluation of 18 studies emphasizing experimental or model-oriented solutions; A middleware concept for IoT transactions in smart cities, focusing on secure IoT node activation using smart contracts and Blockchain technology.
Present study	Analysis of the smart city middleware landscape.	Smart city domains; Enabling technologies; Functional and non-functional requirements; Architectural styles; Programming paradigms.	Interoperability & standardization; Scalability; Security & privacy; Context management; Big Data storage; Resilience & fault tolerance; Quality of Service; Energy efficiency; Evolution & maintenance.	An evaluation of 20 middleware solutions; A conceptual framework to guide future development and research in the domain of smart city middleware.

execute specific tasks and are contingent upon the software and hardware attributes of the affiliated device. This framework specifically integrates the aforementioned agents to augment the service dissemination and resource-sharing capabilities of data sources, making it particularly tailored for service-centric CPS within the smart city context. The efficacy of the proposed framework is assessed through various metrics, including resource and storage utilization, as well as

response time and downtime across diverse query requests and durations, in a simulation environment.

B. CityPulse

The CityPulse framework facilitates the creation of smart city services through a distributed system designed for semantic discovery, data analytics, and interpretation of vast real-time IoT and social media data streams [28]. CityPulse aims to enable cross-domain data integration by not only integrating diverse, uncertain, and incomplete data for reliable information but also offering advanced data analytics modules for intelligent data aggregation, event detection, quality evaluation, contextual filtering, and decision support. The CityPulse framework consists of two main sets of components: large-scale data stream processing modules for interacting with diverse data sources and adaptive decision support modules that provide context-based recommendations. CityPulse applications employ cloud-based components for service execution, allowing for continuous data stream monitoring and processing, and enabling applications to retrieve real-time city status information via APIs. Conversely, the adaptive decision support modules are activated only when specific recommendations or context monitoring is required, distinguishing them from the constantly running data processing tools. The CityPulse middleware components are versatile, suitable for various application domains, and are offered as open-source. To streamline the development process, each CityPulse component furnishes a set of APIs. The framework is exemplified through a Smart Travel Planner application.

C. CIVITAS

Civitas is a distributed object-oriented middleware tailored for smart cities [29]. Civitas provides a range of services, from sensor deployment to high-performance data analysis, with reasoning capabilities that reduce the need for hard-coded elements and enable seamless adaptation to various city deployments. Civitas exhibits several core design principles and characteristics that address challenges such as the inappropriateness of generic middleware for resource-limited devices, resource-intensive communication protocols, restricted flexibility and real-time platform management, the absence of embedded intelligence, and the lack of a development paradigm. The Civitas framework offers high-level services that can be interlinked to form new functionalities, demonstrated through a case study involving license plate tracking. Leveraging a reasoning engine that merges city-specific data with common-sense knowledge (like the law of inertia), the system can predict and track car movements across a network of video cameras, recalculating positions based on last known data and city layout, without relying on hardcoded operations.

D. FIWARE

FIWARE is an open-source platform that provides a set of APIs and tools for the development of smart applications across different fields within the smart city domain [30]. The FIWARE platform offers components for various functionalities such as context data management, security, and interfaces to the IoT, allowing real-time access to sensor data, control of IoT devices, data analytics, and more. NGSI (Next Generation Service Interface) [31] is a critical component of FIWARE for managing context information, enabling applications and devices to update, query, and subscribe to changes in a standardized manner, ensuring interoperability and scalability. NGSI-LD is its evolution [32], providing a modern approach to managing context information by incorporating linked data capabilities. This allows developers to create applications that can interact with the physical world through IoT devices easily and efficiently. FIWARE plays a crucial role in the federation of large-scale pilot sites by providing the necessary infrastructure to enable transparent routing of IoT datasets and streams from providers to consumers. This ensures seamless communication and interoperability among various IoT devices and platforms involved in the federation, ultimately enabling the successful deployment and operation of large-scale IoT applications across multiple sites.

E. FogFlow

FogFlow is a novel fog computing framework based on a standard-driven methodology tailored for IoT smart city platforms [33]. The programming model in FogFlow simplifies the creation of elastic IoT services across both cloud and edge environments and offers standard interfaces for contextual data sharing and reuse across services. It is built on top of FIWARE and adopts the open mobile alliance's NGSI [31], to ensure openness and interoperability in IoT and smart cities. The FogFlow framework, designed for geo-distributed infrastructure resources, connects with users and external applications via its API and interfaces, categorizing infrastructure into cloud, edge nodes, and devices, each serving distinct computational tasks. The framework consists of three divisions: service management, data processing, and context management. It includes specific components like task designer, topology master, and IoT brokers. These components facilitate seamless IoT service orchestration, contextual data management, and cross-domain communication. To explain the framework's applicability in smart city contexts, an example application was implemented centered on anomaly detection in urban energy consumption. Additionally, a performance assessment of FogFlow was provided, drawing from microbenchmarking results that gauge message propagation latency, throughput, and system scalability.

F. GAMBAS

The GAMBAS middleware aims to streamline the development of smart city applications by offering a Java-based runtime system and an accompanying Software Development Kit (SDK) [34]. GAMBAS, through its system components, addresses three primary challenges confronted by developers: efficient data acquisition, secure data distribution

with privacy preservation, and interoperable data integration. To illustrate the utility of this runtime system and SDK, two sample applications are presented, showcasing diverse middleware functionalities. The GAMBAS Voiceprint Launcher is an Android application that allows users to launch apps via trained voice commands by creating and matching audio fingerprints, demonstrating significant middleware functionalities while operating locally without showcasing remote connectivity or certain integration aspects. The Linked Weather application emphasizes data management, remote communication, and the J2SE SDK, integrating with a weather service for major German cities and storing the data in a semantic data storage for accessibility to other devices.

G. InterSCity

InterSCity is an open-source smart city platform founded on a microservice architecture [35]. Its primary objective is to facilitate collaborative research, development, and deployment endeavors in the realm of innovative smart city solutions. The InterSCity platform offers key features across domains like public transportation and environmental monitoring, while delivering high-level cloud services for managing IoT resources and data processing. InterSCity leverages microservice methodology to create a flexible, expandable, and loosely interconnected architecture, designed to adapt to diverse smart city projects. Key principles include achieving modularity through single-purpose services, employing decentralized models with each service having its own evolving database, and promoting decentralized evolution where services operate autonomously with defined boundaries. The platform emphasizes the reuse of open-source projects, the adoption of open standards to prevent vendor lock-in, the prioritization of asynchronous messaging over synchronous interactions, and the endorsement of stateless services to reinforce scalability and flexibility. Experimental results demonstrate the platform's scalability and performance in smart city contexts, with microservices deployable independently.

H. LinkSmart

The LinkSmart middleware, an open-source, service-oriented solution for smart buildings, is further developed into a middleware specifically designed for energy-efficient smart open spaces [36]. LinkSmart equips developers with a suite of components, referred to as managers, designed according to the principles of service-oriented architecture. Each manager manifests specific functions as a web service, facilitating the development of LinkSmart applications and prototypes for various application scenarios. The LinkSmart middleware for smart energy-efficient spaces extends features like secure communication, which allows direct communication among devices regardless of network boundaries; event-based architecture, essential for developing systems and applications in sensor-rich environments; and proxy concept, which abstracts low-level technologies to web services, enabling their transparent use by other LinkSmart components. This service-oriented approach provides flexibility in planning deployments, either in a centralized or distributed manner, addressing key requirements like message encryption, trust management, and the development of loosely coupled event-based systems.

I. MiSCi

MiSCi (Autonomic Reflective Middleware for Smart Cities) is an architecture for making context-aware smart decisions in a smart city [37]. The MiSCi employs a multi-layered architecture grounded in the Multiagent System (MAS) paradigm, inheriting attributes such as sociability, adaptability, and intelligence. This MAS structure comprises agents representing individuals, devices, and applications within a smart city, facilitating inter-service provision. MiSCi employs intelligent agents rooted in web services, utilizing emerging ontologies to adapt to a city's dynamics and address citizens' needs based on real-time context. The basic layers in the MiSCi architecture include: the MAS Management Layer for agent coexistence and interoperability; the Service Management Layer enabling integration of MAS and SOA paradigms, facilitating cloud-based web service interactions; and the Context-Awareness Layer focused on managing context information. Other layers are defined for addressing ontological emergence, logical and physical management of smart city components, and the actual physical layout of the smart city.

J. OpenIoT

OpenIoT is a middleware platform designed to semantically unify a wide array of IoT applications within the cloud environment [38]. Central to OpenIoT is the utilization of the W3C Semantic Sensor Networks (SSN) ontology [39], establishing a standard-based model for representing both physical and virtual sensors. OpenIoT encompasses sensor middleware, facilitating seamless data collection from diverse sensors and ensuring appropriate semantic annotations. It also offers an array of visual tools for IoT application development and claims a capability for mobile sensor integration, catering to the surge in mobile crowd sensing applications. OpenIoT employs a publish/subscribe mechanism to support the discovery and collection of data from mobile sensors, integrating a cloud-based processing engine with a mobile broker for data acquisition. This system allows for efficient pre-filtering near data sources, ensuring only pertinent data is uploaded to the cloud, while also optimizing cloud resource usage by processing multiple subscriptions concurrently, adapting to varying load conditions. OpenIoT offers an Integrated Development Environment (IDE) that features visual tools for defining IoT services, discovering sensors, configuring sensor metadata, monitoring IoT service status, and visualizing services using Web2.0 mashups, thereby streamlining the IoT application development process.

K. RIMWARE

Rimware is a middleware that operates on a service-based model, similar to many other middleware solutions [40].

Its developers have pinpointed two primary challenges addressed by the technology: first, ensuring secure communication between devices and the Cloud, leveraging alternate gateways when main devices like smartphones or set-top boxes are unavailable; and second, the imperative for the Cloud to precisely model the capabilities of connected devices. To assess the efficacy of Rimware in enabling cross-interoperability across diverse applications, a version named Blue Rim was introduced. A scalable and extensible IoT gateway architecture for smart cities was presented, supporting various wireless sensor networks and communication protocols through a modular design. This design comprises independent, easily interchangeable modules for communication, data processing, security, and services. Experimental evaluations validate its scalability, capacity to manage numerous devices and sensors, and adaptability to new networks and protocols.

L. S2NetM

Semantic Social Network of Things Middleware (S2NetM), is a smart city middleware designed to capitalize on social relationships for reinforcing semantic interoperability within IoT-based environments [41]. The S2NetM utilizes semantic reasoning and alignment methodologies, promoting the development of adaptive, context-driven social networks of entities. These networks synergistically function, paving the way for innovative IoT-centric smart city applications. S2NetM provides robust solutions to the semantic interoperability challenge, harnessing sophisticated ontology and reasoning mechanisms to curate a uniform communication framework and data schema. This facilitates coherent data interchange and substantive engagement amongst varied types of IoT devices in social settings. Moreover, S2NetM integrates features such as dynamic relationship selection, trustworthiness management, and streamlined service discovery, addressing the dynamic requirements of social IoT contexts. In concert, these elements underpin sophisticated smart city applications. An assessment utilizing a tangible use case underscores the middleware's efficacy and applicability.

M. SEDIA

SEDIA serves as a platform tailored for the development of smart city applications, harnessing a variety of data sources, including geographical information [42]. This platform leverages a semantically enhanced data model, pivotal for effective data analysis and seamless integration. SEDIA platform collects, stores, and processes data, using diverse IoT technologies and communication infrastructure. A pivotal phase in data integration involves acquiring, structuring, and preparing data for semantic annotation. Employing semantic tools and techniques, advanced data analysis is executed, extracting insights to facilitate the inception of novel functions and services in the application layer. The ''Green Route'' application, developed as a proof-of-concept, guides users on the shortest path with optimal air quality. It achieves this by collecting data from various IoT devices and open platforms, then semantically enriching and harmonizing this data. The processed data is stored in a Neo4j graph database, aiding in identifying pollution patterns. Finally, a web application visualizes this data, enabling users to identify low pollution walking routes.

N. SGeoL

The Smart Geo Layers (SGeoL) serves as a platform tailored for the development of smart city applications [18]. In addition to integrating urban data with geographic information, SGeoL encompasses advanced abstractions for aspects such as context data management, the consolidation of diverse data, semantic support, data examination and representation, and provisions for data security and privacy. SGeoL utilizes components from the FIWARE platform as its foundational middleware infrastructure [30]. Although FIWARE does not possess a multi-domain data model centered on city geography for advanced semantic analysis [43], the integration of FIWARE components in SGeoL serves to simplify development tasks and introduce additional valuable services. The proposed middleware architecture is designed to address several challenges associated with smart city applications, including high network latency, limited network bandwidth, and high data volume. The proposed middleware architecture is evaluated using a set of experiments, which demonstrate that it is capable of processing large amounts of data with low latency and high efficiency.

O. SMArc

SMArc (Semantic Middleware Architecture) is a middleware solution for smart city energy management [44]. The objective is to process the gathered data, abstracting applications from the intricacies of metering facilities, and ensuring that any modifications at these foundational levels are incorporated for subsequent system operations. SMArc integrates features including accommodating low capability devices, prioritizing security and privacy, functioning as a distributed system, addressing specific Smart Grid challenges, incorporating semantic features with a lightweight ontology, and leveraging an inference engine for proactive actions based on data interpretation. SMArc is grounded on a semantic middleware architecture, which includes modules for ontology integration, semantic data storage, service management, hardware resource management, and semantic information processing. It acts as an intermediary between the application and communications layers. SMArc tackles middleware challenges such as interoperability, scalability, and heterogeneity by uniquely incorporating semantics into its design and introducing an inference engine for decision-making.

P. SmartCityWare

SmartCityWare is a service-oriented middleware designed to integrate and leverage the Cloud of Things (CoT) and fog computing, providing a suite of services to support smart

city applications [45]. This middleware conceptualizes all system resources as a set of services, which facilitates the development of smart city applications. A key advantage of this approach is the flexibility it offers in extending the middleware to incorporate new and advanced services as smart city applications evolve. Broker, invocation, location-based and security services are among the fundamental services in SmartCityWare. These services are crucial for maximizing the effectiveness of other available services and for enabling the primary functions offered by the middleware. SmartCityWare includes a customized multi-agent infrastructure, developed for heterogeneous systems and modified to support the service oriented computing model of SmartCityWare, which enables it to handle heterogeneous environments composed of fog, cloud, and IoT devices.

Q. SmartSantander

The SmartSantander project focuses on establishing a unique European test facility for IoT research and experimentation within a smart city context, emphasizing the significance of real-world conditions, infrastructural scale, and diverse application domains in dense techno-social ecosystems [46]. The proposed reference model for IoT experimentation testbeds consists of both testbed observation/management and IoT experimentation layers, divided into four main subsystems: Authentication, Authorisation, and Accounting (AAA); Testbed Management; Experimental Support; and Application Support. Each subsystem, spanning across different node-tiers, is made up of functional blocks that offer specific functionalities through a range of APIs which can manifest as Web Services, RESTful APIs, etc. Detailed descriptions of each subsystem reveal their distinct functionalities including accessing AAA controls, providing automatic facility management, supporting users throughout the experimentation life-cycle, and providing data management functions for various applications, including smart city services and experimental data access.

R. Snap4City

Snap4City is an open-source IoT platform which supports the development of advanced IoT applications through urban dashboards and mobile interfaces [47]. According to the Snap4City approach, the adoption of IoT applications in smart cities allows individual users to manage and configure their applications, necessitating a controlled backend system to prevent potential issues from intentional misuse, non-expert errors, or redundant processes. This requires the development of tools for real-time bandwidth monitoring and periodic evaluations of message handling capacities on cloud-based IoT application execution. The platform also aims to address several non-functional requirements, including: openness, scalability, adherence to standards, robustness, management of heterogeneous communications, interoperability, and commitment to security and privacy standards. The Snap4City platform is designed with a suite of components that enable data collection from diverse sources, its storage and management in knowledge bases and noSQL storages, and the development of IoT applications using microservices. It further facilitates the creation of data analytics and transformation services, the execution of smart city processes on cloud infrastructure, the visualization of data via city dashboards, and provides access through specialized microservices and advanced APIs for web and mobile applications.

S. SOUL

SOUL middleware employs stream reasoning technology to deliver services like fire accident management, necessitating real-time processing and intricate reasoning [48]. SOUL provides a stream reasoning system model tailored for smart city applications, embedded within the smart city middleware, leveraging real-time big data processing technologies. The system employs Apache Kafka for message processing and Apache Storm for real-time distributed processing to address these real-time constraints. SOUL adeptly processes data from the smart city infrastructure and offers services via the city portal tier, comprising layers like the Common Device Interface, which receives data from various sensors and transfers it to the Smart Computing Layer, ensuring compatibility with heterogeneous Ubiquitous Sensor Networks. The Smart Computing Layer, pivotal for stream reasoning, incorporates components like the Context Converter, which translates raw sensor data into RDF/OWL format suitable for stream reasoning, and the Context Analyzer, which performs reasoning using predefined rules and leverages cloud computing for enhanced performance.

T. WeValue

WeValue is a Blockchain-supported platform designed to enhance societal value exchange and co-creation in smart cities by facilitating spontaneous interactions among various stakeholders, primarily citizens, and aligning complementary or similar interests to promote neighborhood-level value co-creation processes [49]. The approach in WeValue is centered on two core concepts: Value Agents and Value Contracts. A Value Agent is a software agent that actively represents a stakeholder, participating in task matching and orchestration processes to identify, negotiate, and approve suitable task compositions. A Value Contract is a Blockchain-based smart contract that formally specifies the agreement for a collaborative activity, encoding contractual obligations, rewards, and penalties. It is invoked by agents to verifiably fulfill obligations and determine rewards, using agreed oracles and incentivizing mechanisms to facilitate automated negotiation and successful conclusion among Value Agents. The platform utilizes the SmartSociety platform's APIs [50] for collaborative computing, enabling Value Agents to manage tasks, communicate with users, and manage privacy and rewards, while also incorporating modified components to ensure seamless integration, privacy, communication, and incentive administration in accordance with Value Contracts.

III. ENABLING TECHNOLOGIES

The most pivotal enabling technologies used in smart city platforms are discussed in the following sections. Not only do these technologies play a foundational role in the overall functioning and optimization of these platforms, but they also influence the operation, adaptability, and efficiency of smart city middleware.

A. INTERNET OF THINGS

IoT devices, such as sensors and actuators, are instrumental in collecting and transmitting data from various city domains [12]. This data can then be analyzed and acted upon, making IoT a crucial component of smart city middleware. In this context, the middleware's role is multifold. It serves as the critical bridge between these IoT devices, which rely on M2M communication techniques, and the smart city applications that process and act upon the data [51]. Middleware manages device connectivity, ensures secure and reliable data transmission, handles data formatting and normalization, and supports real-time processing and analysis. Leveraging M2M protocols, the middleware facilitates seamless communication between devices, enhancing the dynamism and responsiveness of the system. Furthermore, it enables the orchestration of actuators based on the derived insights. The middleware is therefore not only crucial for managing the technical aspects of device connectivity and data management, but it also plays a fundamental role in transforming the raw data into actionable insights and orchestrating the execution of these insights in the real world.

B. DISTRIBUTED DATA PROCESSING AND MANAGEMENT

Smart city infrastructures are inherently heterogeneous and deal with dynamic data flows, making centralized platforms inadequate due to scalability and latency challenges [52]. The growing shift towards distributed data processing allocates tasks across interconnected tiers, each vital for timely, efficient, and secure data processing in urban environments [53]. Figure 1 provides a conceptual overview of the distributed data processing and management within smart city middleware. The tiers discussed in subsequent sections each bring unique advantages. Together, they form a robust, scalable, and resilient framework tailored for the diverse needs of smart city ecosystems. Within this framework, middleware serves as the cohesive element, facilitating interoperability and the smooth transition of data and control commands across various smart city applications.

1) CLOUD COMPUTING

Cloud platforms provide the necessary computational resources and storage capabilities required for handling, processing, and storing the massive volumes of data generated across various sectors of a smart city [54]. By combining the strengths of both IoT and cloud computing, the Cloud of Things model has emerged as a comprehensive solution that can be used to drive intelligent decision-making and actions in the smart city applications [45], [55]. Extensions of cloud computing models tailored for the smart city context have been also proposed using the terms City Application Software as a Service (CSaaS) and City Platform as a Service (CPaaS) [15]. The CSaaS model allows city administrations to access sophisticated applications without having to develop, manage, or maintain them in-house. CPaaS, on the other hand, refers to a cloud environment that provides a platform for developing, running, and managing city applications. This accelerates the development process, reduces the time-to-market for new applications, and makes it easier to iterate and improve existing applications. Both these models allow for seamless integration of diverse city applications, enable the efficient use of resources, and allow for the rapid adoption of innovative solutions. This plays a crucial role in the realization of smart city goals and makes cloud computing an essential component of smart city middleware as evidenced by the vast majority of the solutions examined.

2) EDGE AND FOG COMPUTING

Edge and fog computing have emerged as pivotal solutions for addressing latency-sensitive applications and alleviating network infrastructure burdens in smart cities [56]. By situating computation closer to data sources, such as IoT devices, edge computing optimizes latency and bandwidth. Fog computing, bridging the gap between edge and cloud computing, allocates computation to local nodes like gateways, striking a balance between proximity to data sources and computational power. This strategic positioning enables prompt processing, essential for many smart city tasks [57]. This computing model has been adopted by FogFlow [33] to enable smart city IoT services, SEDIA [42] to ensure swift data collection and translation despite computational and storage constraints, SmartCityWare [45] for efficient resource utilization, and SmartSantander [46] to provide a programmable experimentation substrate.

3) MOBILE CROWD SENSING AND COMPUTING

Mobile Crowd Sensing and Computing (MCSC) taps into the sensing capabilities of prevalent mobile devices like smartphones, wearables, and in-vehicle systems to gather and analyze urban data [58]. This approach, when integrated into smart city middleware, complements data from traditional sources, offering a more detailed view of urban dynamics [59]. For example, OpenIoT provides the ''Urban Crowdsensing Service'' so that volunteers with wearable sensors can provide real-time air quality data [38]. Similarly, the S2NetM solution merges crowd-sourced data with inputs from IoT devices, forming a foundation for its decision-making [41]. The relevant approach in SmartSantander is also noteworthy; it transforms mobile phones from mere communication tools to versatile sensors that capture diverse data, from locations to environmental conditions [46]. This platform even allows users to receive alerts about ongoing city events, fostering a deeper connection with their urban

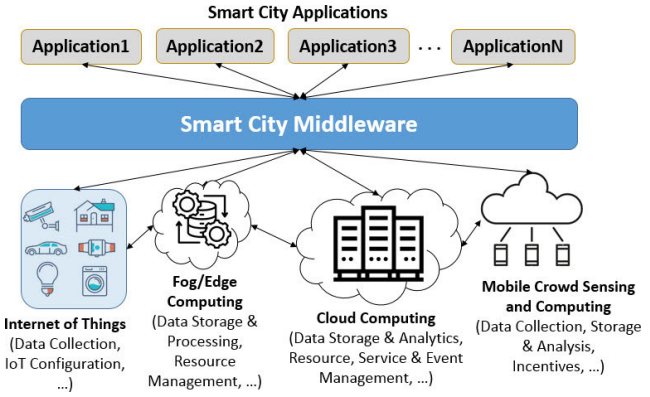

FIGURE 1. Distributed data processing and management in the context of smart city middleware.

surroundings. While there is a superficial similarity in that both MCSC and edge computing can involve local data processing on devices, their objectives, challenges, and primary use cases are quite distinct [60]. MCSC emphasizes crowd-sourced data collection from diverse user devices, addressing challenges related to incentivization, data reliability, and user privacy, while edge computing prioritizes processing data near its source and is more concerned with ensuring data integrity, reducing transmission delays, and safeguarding local processing nodes. Additionally, emerging paradigms like Dew/Mist Computing are extending the concept of MCSC by not only utilizing mobile devices for data collection but also leveraging their computational power. In this model, smartphones and similar devices act as nodes for local data processing, decentralizing the computational load and enabling more efficient, sustainable handling of urban data [61], [62].

C. BIG DATA

In the context of smart cities, Big Data refers to the vast amounts of structured and unstructured data generated from various sources, including IoT devices, sensors, social media, public services, and many others [63]. The data in this context is distinguishable by its inherent attributes, namely its volume, velocity, variety, veracity, and value [64]. Smart city middleware is essentially the technological glue that binds together all these data sources and enables the processing and analysis of this information [28], [38], [45], [65]. It is responsible for handling data integration, heterogeneity, security, privacy, and real-time processing, among other tasks. Big Data technologies, such as Hadoop, Spark, and NoSQL databases, among others, are often integrated into the middleware to efficiently process and analyze the collected data [18], [47], [66]. An additional critical element concerning data within smart cities is the notion of Open Data [67]. In a smart city context, this could encompass data on various city services such as public transportation schedules, air quality indices, city budget allocations, public project statuses, and more. Open Data initiatives in smart cities can significantly enhance the diversity of Big Data available for analysis [68]. Such openness and transparency can spur the development of innovative applications and services that elevate urban living, as evidenced by several of the middleware solutions examined [18], [28], [30], [42], [47].

D. ARTIFICIAL INTELLIGENCE

In the realm of smart city middleware solutions, there is a growing emphasis on harnessing Artificial Intelligence (AI) as an enabling technology [69]. AI, particularly machine learning and deep learning techniques, can process and analyze large datasets, identify patterns, make predictions, and even make decisions [70], [71]. For, instance, CityPulse incorporates a quality monitoring module that utilizes machine learning to evaluate the data quality from various input sources [28]. Also, the MiSCi middleware embeds AI functionalities, demonstrating its capacity for learning, autonomy, and reasoning, thereby ensuring that the city's digital responses are optimized based on past experiences and data [37]. OpenIoT expands on this technology by incorporating machine learning algorithms and statistical methods, facilitating enhanced data processing and informed decision-making [38]. Furthermore, the server tier in SmartSantander offers a platform for exploring real-world data mining and knowledge engineering techniques for optimizing resource allocation and facilitating proactive decision-making [46].

E. CYBER-PHYSICAL SYSTEMS

Cyber-Physical Systems (CPS) are integrations of computation, networking, and physical processes [72]. In the context of a smart city, CPS can include systems like smart grids, autonomous vehicle systems, medical monitoring, process control systems, distributed robotics, and automatic pilot avionics [73]. An emerging perspective in this domain is the concept of 'Digital Twins', which are virtual representations of physical assets, systems, or processes [74], [75]. Extending beyond smart cities, the paradigm of CPS is also fundamental to Industry 4.0, characterized by interconnected systems, automation, and data exchange in manufacturing technologies [76]. Middleware for a smart city must bridge the cyber world of computing and the Internet with the physical world of infrastructure and systems. In this context, CPS offers a model for such integration, as exemplified by AMF-CPS [27]. InterSCity middleware, on the other hand, facilitates the development of applications such as smart parking infrastructures, underpinned by CPS [35]. OpenIoT stands as a platform that fosters the integration and seamless management of CPS, facilitating data collection and processing across both virtual and tangible devices [38]. Finally, SmartCityWare conceptualizes the components of smart cities in both cyber and physical worlds as service providers, ensuring that application developers can harness these resources without being encumbered by their inherent heterogeneity [45].

F. CYBERSECURITY

Given the critical nature of data in smart cities, robust cybersecurity measures, such as encryption, intrusion detection

systems, and blockchain technology, are indispensable [49], [77]. Such measures not only ensure data confidentiality, integrity, and availability but also protect against unauthorized access, alterations, or service interruptions [34], [38], [78]. Platforms like SGeoL [18] and Civitas [29] exemplify this commitment by managing user authentication, security policies, data protection, and ensuring device certification respectively. Infrastructure safety is paramount, as potential cyber-attacks could result in physical harm or service disruptions [45], [79]. The security of IoT devices is equally critical, especially given the potential for breaches [80]. As data seamlessly integrates across smart city systems, secure communication, such as the use of secure MQTT with TLS in SEDIA [42], becomes vital. Moreover, the focus of cybersecurity transcends prevention, emphasizing resilience under adversities and swift recovery post incidents [46], [81]. With governments setting cybersecurity standards, especially for pivotal sectors, middleware must adhere to these regulations to ensure urban system safety and evade legal consequences [49], [82]. Cybersecurity, therefore, emerges as a foundational enabling technology for smart city middleware, ensuring that the digital backbone of urban systems remains robust, resilient, and trustworthy.

G. SEMANTIC WEB AND ONTOLOGIES

The integration of Semantic Web and Ontologies is a prominent feature across several smart city middleware solutions, enhancing data representation, validation, and query capabilities [18], [29], [30], [38], [40]. Semantic Web technologies, including RDF (Resource Description Framework) and OWL (Web Ontology Language), offer standardized ways to represent and share data, facilitating interoperability between different smart city applications and platforms [19]. The ontology in S2NetM showcases a unique representation of social relationships among IoT devices and incorporates ontology alignment and reasoning mechanisms for efficient data querying [41]. Both SGeoL [18] and SEDIA [42] harness Semantic Web technologies for data annotation, validation, and inferencing, promoting advanced data analysis and semantic searches. Semantic Web technologies, especially ontologies, provide a standardized way to represent and integrate diverse data, ensuring that different systems can understand and work with it cohesively [83]. For instance, Snap4city utilizes the Km4City Ontology to aid in representing diverse city data, while also facilitating semantic discovery of sensors and actuators [47]. Furthermore, middleware equipped with ontology-based reasoning capabilities can make inferences based on existing data [18], [41], [42]. By understanding user preferences and behaviors through semantic annotations, middleware can provide more personalized services. For instance, if a user prefers ''green spaces'' and ''quiet areas'', the middleware, as exemplified by SEDIA ''Green Route'' use case [42], can recommend routes or destinations that fit these criteria.

H. BLOCKCHAIN

Blockchain provides a secure, transparent, and decentralized framework, enhancing the efficiency, trustworthiness, and automation of processes in smart cities [84], [85]. Its cryptographic mechanisms ensure secure, peer-to-peer transactions [86], while smart contracts, such as those central to the WeValue platform functionality [49], automate and enforce agreements, eliminating the need for intermediaries [87]. Beyond security, Blockchain's immutable identity verification is crucial for personalized city services and secure access [88], and its traceability ensures the authenticity of data and transactions [89]. In smart cities, this can be useful for tracking supply chains, verifying the source of goods, or ensuring the accuracy and origin of environmental data. Beyond strengthening cybersecurity, Blockchain has the potential to transform data storage, transaction handling, and service delivery in smart city middleware through its decentralized and transparent capabilities.

I. SUMMARY

Table 2 summarizes the enabling technologies adopted by the examined smart city middleware solutions, offering a comprehensive overview of the prevalence and integration of these technologies in the context of urban digital infrastructure. A review of these middleware solutions points out the ubiquity and significance of technologies such as the IoT and Cloud Computing, which are almost universally adopted across the solutions. Cybersecurity and Semantic Web and Ontologies technologies are similarly recurrent, reflecting the need for robust data protection and meaningful data interpretation. Big Data and AI, vital for scalable data management and intelligent decision-making, are also noticeably present in several middleware platforms. Meanwhile, Fog/Edge computing or MCSC are harnessed by a subset of these solutions, emphasizing their relevance in particular smart city scenarios. Finally, the presence of CPS and Blockchain in a few solutions gives their potential in enhancing seamless interactions between the digital and physical worlds as well as transparency and security. This distribution highlights the diverse requirements and objectives of smart city middleware, emphasizing the need for a multifaceted approach to address the complex challenges of smart cities.

IV. SMART CITY MIDDLEWARE REQUIREMENTS

To address the research question regarding essential requirements for smart city middleware, this section examines the functional and non-functional requirements observed in the studied middleware platforms. The analysis is based on two primary criteria: direct references in literature affirming a platform's implementation of a requirement, and the evident presence of components within the platform that meet the specified requirement. This dual-focused approach aims to provide a holistic understanding of what a smart city middleware should include.

TABLE 2. Enabling technologies used by the smart city middleware solutions.

Smart City Middleware	Internet of Things	Cloud Computing	Edge & Fog Computing	Mobile Crowd Sensing	Big Data	Artificial Intelligence	Cyber-Physical Systems	Cybersecurity	Semantic Web & Ontologies	Blockchain
AMF-CPS	√	√	×	×	×	×	√	×	×	×
CityPulse	√	√	×	×	√	√	×	×	√	×
Civitas	×	√	×	×	×	√	×	√	×	×
FIWARE	√	√	×	×	√	×	×	√	√	×
FogFlow	√	√	√	×	×	×	×	×	×	×
GAMBAS	×	√	×	×	×	×	×	√	√	×
InterSCity	√	√	×	×	×	×	√	×	×	×
LinkSmart	√	√	×	×	×	×	×	×	√	×
MiSCi	×	√	×	×	×	√	×	×	√	×
OpenIoT	√	√	×	√	√	√	√	√	√	×
Rimware	√	√	×	×	×	×	×	√	×	×
S2NetM	√	√	×	√	×	×	×	√	√	×
SEDIA	√	√	√	×	×	×	×	√	√	×
SGeoL	√	√	×	×	√	×	×	√	√	×
SMArc	√	×	×	×	×	×	×	√	√	×
SmartCityWare	√	√	√	×	√	√	√	√	×	×
SmartSantander	√	√	√	√	×	√	×	√	×	×
Snap4City	√	√	×	×	√	×	×	√	√	×
SOUL	√	√	×	×	√	×	×	×	√	×
WeValue	×	×	×	×	×	×	×	√	×	√
Count	**16**	**18**	**4**	**3**	**7**	**6**	**4**	**13**	**12**	**1**

A. FUNCTIONAL REQUIREMENTS

1) DATA MANAGEMENT

a: DATA COLLECTION AND INTEGRATION

Middleware should be capable of collecting data from numerous and heterogeneous sources, including IoT devices, sensors, databases, and third-party systems [30], [34], [38], [48]. To enable seamless integration from third-party sources, the middleware can employ REST APIs [28], [35], [41], embrace the publish/subscribe paradigm [33], [36], [46] and leverage open data platforms [42], [47]. In essence, data collection and integration capabilities empower smart city middleware to bridge the gap between raw data and actionable insights. Integration, in this context, is not merely about accumulating data; it is about harmonizing and translating this data into a cohesive format, suitable for further analysis and action. As an illustration, the Data Integrator component in SGeoL integrates data from multiple external systems, standardizing it for accessibility via the Data API [18]. Given the heterogeneity of sources, this process also entails data validation, ensuring that the incoming data is consistent and reliable [45], [46].

b: DATA STORAGE AND RETRIEVAL

Middleware in smart cities must provide robust data storage solutions with efficient retrieval and querying capabilities [33], [45]. It should accommodate both structured data, like that in relational databases, and unstructured data, such as videos or social media content [18], [42], [46]. As smart city data needs grow in complexity, middleware should adapt to advanced storage methods, including distributed databases and cloud architectures [18]. Given the demand for real-time data in many smart city applications, the middleware must also efficiently process complex queries from multiple sources simultaneously, ensuring minimal latency [28], [33]. More specifically, SEDIA uses the Neo4j graph database to store semantically annotated data and to identify patterns and relationships during retrieval [42]. SGeoL integrates multiple databases, including MongoDB, Apache Jena, and PostgreSQL with PostGIS, to handle diverse data types [18]. In contrast, SMArc focuses on semantic data storage within its Repository Module [44], while Snap4City employs a knowledge base and noSQL storages for efficient data retrieval with advanced search functionalities [47]. Finally, WeValue maintains a centralized data-store for user and group profiles, emphasizing secure storage for sensitive data, aiding Value Agents in negotiations [49].

c: DATA PROCESSING

Data processing emerges as a key requirement in smart city middleware solutions, pivotal for transforming raw data into actionable intelligence. Such middleware not only

emphasizes tasks like resource management, as seen in AMF-CPS [27], or real-time pattern detection, as in FIWARE [30], but also navigates the intricacies of data processing [38]. Efficient filtering mechanisms are vital, separating essential information from the overwhelming volume of incoming data. This selective approach, demonstrated by platforms like GAMBAS [34] and LinkSmart [36], ensures optimal use of computational resources. Beyond mere normalization, middleware often undertakes intricate transformations, converting textual data into numerical formats or reshaping datasets for specific analytical paradigms. SEDIA [42] and SGeoL [18], for instance, emphasize varied processing tasks, from intensive analysis to data aggregation. Middleware should also support data fusion, merging multiple streams into comprehensive views. As exemplified by WeValue [49] and SmartCityWare [45], merging data, especially when coupled with semantic enrichment or compliance adherence, amplifies its value by offering a richer view of information. Whether it is the S2NetM focus on dynamic SIoT environments [41] or the OpenIoT versatility in data processing modes [38], it is clear that the ability to efficiently process, transform, and fuse data is central to the adaptability and efficacy of smart city middleware solutions.

2) ANALYTICS AND REPORTING

Middleware must offer a robust suite of analytics, visualization, and reporting tools to transform the massive data generated in smart cities into meaningful insights, patterns, and trends for informed decision-making. Visualization aids in converting intricate data into comprehensible formats, while customizable reporting, coupled with predictive analytics, enables administrators to understand current urban conditions and anticipate future challenges or opportunities as demonstrated in the Smart City Magnifier use case in FogFlow [33]. Similarly, LinkSmart emphasizes analytics through occupancy data analysis and underscores the significance of detailed reporting and visualization, particularly concerning environmental conditions and energy usage [36]. On the other hand, CityPulse integrates diverse data and uses advanced analytics in its decision support modules to offer real-time, context-aware city recommendations [28]. SGeoL harnesses a Complex Event Processing Engine for its data analysis tasks [18]. SmartCityWare fosters collaborative analytics across smart buildings, aiming for precise models for applications ranging from fault detection to demand optimization [45]. Snap4City provides a range of tools for data analytics in environments like R, Java, and Python [47].

3) IoT MANAGEMENT

IoT management in smart city middleware is imperative for the integration and operation of a plethora of internet-connected devices, which includes seamless onboarding, remote configuration, continuous monitoring, lifecycle management, and stringent security protocols. The following provides examples of such implementations. FIWARE incorporates multiple IoT agents tailored for diverse protocols, streamlining the management of IoT devices [30]. LinkSmart offers capabilities such as remote reconfiguration of sensor node parameters, enhancing adaptability [36]. Rimware, designed to bolster IoT-Cloud integration, demonstrates its focus on IoT management through case studies on BLE device interactions and multi-adapter gateway functionalities [40]. SEDIA ensures continuous integration and communication across various IoT devices, middleware, and service layers [42]. The IoT Manager component in SGeoL simplifies device discovery and facilitates their integration using various protocols [18]. Finally, SmartSantander augments its management capabilities by supporting wireless reprogramming of experimentation nodes, ensuring reliability through multipath communication, and rapid malfunctioning node detection [46].

4) SERVICE MANAGEMENT

Middleware for smart cities should be endowed with comprehensive service management capabilities. This encompasses the discovery, orchestration, deployment, and monitoring of diverse services within the smart city ecosystem. CityPulse, for instance, highlights service discovery and composition for effective application deployment [28]. Civitas orchestrates diverse entities, from citizens to public institutions, highlighting the imperative of managing communication among heterogeneous devices and services [29]. Both FIWARE [30] and OpenIoT [38] design prioritize adaptive service management. The Service Management Layer in MiSCi platform facilitates bi-directional integration between agents and cloud services [37], and Rimware emphasizes structured service organization [40]. S2NetM automated service discovery and composition within the SIoT network [41], and WeValue integrated lifecycle management, value agents, and contracts for collaborative tasks [49].

5) CONTEXT MANAGEMENT

In smart city middleware, context management emerges as an essential functional requirement, facilitating real-time adaptability and responsiveness. FIWARE, for example, centralizes this capability with its Orion Context Broker, which manages real-time context information [30]. FogFlow showcases a more intricate design, incorporating federated brokers, orchestrating data flow, caching entity views, and serving context updates across various cloud and edge nodes [33]. The Context-Awareness Layer in MiSCi addresses context discovery, modeling, reasoning, and distribution [37]. S2NetM emphasizes real-time decision-making through its Context Management component [41]. SGeoL collaborates with FIWARE's Orion Context Broker, employing the NGSI-LD protocol and enhancing it with a city model knowledge base [18]. Finally, the Smart Computing Layer in SOUL notably uses the Context Converter and Context Analyzer to process raw data, backed by cloud computing [48].

6) EVENT AND ALERT MANAGEMENT

Middleware should be capable of identifying specific events or patterns within incoming data streams and be equipped to

promptly trigger relevant alerts or initiate appropriate actions in response to such detections. The subsequent examples illustrate how the examined solutions handle this requirement. CityPulse focuses on event detection, crucial for informed decision-making in urban environments [28]. Civitas utilizes event-based communication, optimized through geopositional and time tagging, ensuring precise synchronization and delivery [29]. MiSCi integrates a comprehensive system spanning smart objects, semantic event identification, and predictive monitoring, highlighting the intricate nature of event management [37]. The SEDIA platform employs a combination of coroutines, a Message/Event Broker, and semantic reasoning to efficiently manage events and detect crucial data anomalies [42]. Also the Event Broker in SmartSantander facilitates a distributed 'Event Bus', connecting all testbed management components through a topic-based publish-subscribe model, ensuring asynchronous and distributed event handling [46].

7) RESOURCE MANAGEMENT

Resource management in smart city middleware involves not only the optimal allocation of computational resources but also the efficient management of physical and logical entities that constitute the city's infrastructure. This optimal allocation and efficient management is crucial for ensuring the uninterrupted and efficient performance of all services, ultimately enhancing the overall responsiveness and reliability of smart city systems. AMF-CPS middleware, for example, relies on concurrent and parallel processing by allocating computing and storage resources from a distributed environment to provide in-time responses for user queries [27]. The FogFlow framework vertically partitions infrastructure resources into cloud, edge nodes, and devices [33]. This enables computationally intensive tasks to be processed on cloud servers, while tasks like stream processing are moved to edge nodes; it operates across these geo-distributed, hierarchical, and heterogeneous resources, which may possess either both computation and communication capabilities or only one of them. On the other hand, the InterSCity platform involves the manipulation of city resources, where a city resource is a logical concept that encapsulates a physical entity that makes up the city, such as cars, buses, traffic lights, and lampposts [35]. Similarly, in the context of resource management in CityPulse, a resource is considered a Data wrapper [28].

8) APPLICATION DEVELOPMENT, DEPLOYMENT AND MANAGEMENT

Middleware in smart city ecosystems should provide a robust framework for the seamless development, deployment, and life-cycle management of applications. CityPulse [28], for example, offers component-specific APIs, emphasizing efficient smart city applications, while Civitas [29] and SEDIA [42] present enhanced development environments through interoperability, user-friendly deployment tools, and versatile application toolkits. SGeoL [18] and Snap4City [47] extend this support with integrated dashboards for development, data visualization, and extensive monitoring tools, respectively. The SmartSociety platform, foundational to WeValue [49], advocates an open-source methodology for end-to-end life-cycle management of collaborative initiatives. Beyond just SDKs and developer tools, it is crucial for middleware to ensure agile adaptability, accommodating evolving urban needs without disruptions. This encompasses version control, rollback functionalities, and health monitoring of applications, reinforcing the importance of application development, deployment, and management in maintaining the resilience and adaptability of smart city solutions.

B. NON-FUNCTIONAL REQUIREMENTS

1) SCALABILITY

Scalability emerges as a crucial non-functional requirement for smart city middleware, reflecting the increasing data volume and user dynamics. CityPulse [28], for instance, emphasizes its design to accommodate large-scale data analytics, while FIWARE [30], similarly, is architected with scalability at its core, readying it for the escalating data and service demands typical of smart city scenarios. FogFlow adopts a distributed context management approach, showcasing enhanced scalability [33]. InterSCity achieves scalability through microservices and multi-node deployments [35]. MiSCi blends paradigms like MAS and cloud computing, with simulations attesting to its performance [37]. SEDIA emphasizes scalability, evidenced by its capability under high concurrent loads [42]. SGeoL ensures consistent performance with horizontal scalability and cloud support [18]. SmartCityWare leverages distributed fog nodes to localize tasks [45], and SOUL, using Apache Kafka, guarantees scalable real-time processing [48].

2) INTEROPERABILITY

In the field of smart city middleware, interoperability stands as a keystone feature [19]. Middleware should support various protocols, standards, and data formats to ensure seamless communication and operation between diverse city systems and applications. For instance, CityPulse employs semantically annotated datasets, ensuring machine-readable representations of data streams [28]. FIWARE, on the other hand, focuses on standard APIs, ensuring consistent interoperability among varied services and applications [30]. This emphasis on standardization is further mirrored in FogFlow which adopts the NGSI context management interface [31], a widely accepted open data model. InterSCity [35] and SmartSantander [46] harness standard data models and open communication protocols. Middleware solutions like MiSCi [37], OpenIoT [38], S2NetM [41] and SEDIA [42] emphasize the use of semantic web technologies and ontologies to manage dynamic semantic interoperability or for a unified understanding across devices and applications. Rimware leans on device description at the device level for interoperability [40]. SGeoL employs a unified data model

with semantic support, ensuring comprehensive understanding across systems [18].

3) SECURITY AND PRIVACY

In line with the discussion in subsection III-F on Cybersecurity, the non-functional requirement of Security and Privacy in smart city middleware is of high importance. Middleware solutions that incorporate cybersecurity measures manifest robust components and modules, reflecting the considerations detailed in the aforementioned subsection.

4) ADAPTABILITY

Adaptability in smart city middleware is vital, given the constantly evolving urban context. CityPulse incorporates adaptability through a quality-aware federation of IoT streams tailored for smart city applications [28]. GAMBAS emphasizes adaptability by ensuring that services are responsive to citizens' real-time situations, behaviors, and intents [34]. InterSCity adopts a flexible architecture, allowing customization to meet the specific needs of various smart city initiatives [35]. LinkSmart adapts to user behavior patterns, underscoring its ability to accommodate distinct user behaviors [36]. MiSCi integrates an Autonomic Computing loop and ontological contextual emergence, enabling self-adaptation and real-time service adjustment based on city contexts [37]. OpenIoT employs mechanisms for filtering and annotation, showcasing flexibility with varying data formats [38], while Rimware uses gateway adapters to achieve interface adaptability [40]. Smarc [44] and SmartCityWare [45] both display adaptability through ontology updates and the potential application of learning algorithms, respectively. The diverse strategies employed by the aforementioned middleware solutions highlight the multidimensional nature of adaptability in smart cities, illustrating that there is no universal solution to ensuring responsiveness to smart city application scenarios.

5) CONFIGURABILITY

As demonstrated by the representative middleware solutions discussed below, configurability emerges also as an essential feature. CityPulse offers flexibility by allowing developers to deploy a selective subset of components based on specific application requisites [28]. FIWARE extends a configurable platform, enabling developers to selectively tailor the components they employ [30]. FogFlow introduces a programming model emphasizing configurability through declarative hints, simplifying task configuration for developers [33]. The InterSCity platform employs tools like an automation engine to streamline deployments, highlighting the significance of configurability in deployment procedures [35]. OpenIoT emphasizes user-centric configurability by enabling application development with minimal programming [38]. SmartCityWare [45] and SmartSantander [46] both emphasize the importance of configurable services, ensuring that platforms continually deliver optimal performance.

6) REAL-TIME DATA HANDLING

For many smart city applications, real-time data processing and decision-making are crucial. Thus, the middleware should be capable of handling data streams in real-time. FIWARE is designed with components specifically for real-time data processing and analysis, meeting the pressing demands of smart city contexts [30]. LinkSmart focuses on real-time monitoring and management of energy consumption, emphasizing timely responsiveness to energy usage dynamics [36]. OpenIoT is engineered for swift data processing, with its publish/subscribe engine adeptly handling variable publication rates [38]. SEDIA provides real-time semantic labeling through its Service Layer, ensuring immediate data interpretation [42]. Finally, the Complex Event Processing Engine in SGeoL conducts real-time analyses, promptly identifying intricate events [18].

7) DISTRIBUTED PROCESSING

Distributed processing has emerged as an indispensable feature of modern middleware solutions, given that the majority of them adopt a distributed architecture to cater to the intricate demands of contemporary urban environments. Emerging paradigms, such as edge and fog computing, introduce a hierarchical, decentralized computation model, enabling processing at multiple nodes between the data source and the cloud. These distributed processing models amplify real-time analytics, decision-making, resource efficiency, and system responsiveness. In practice, middleware solutions like AMF-CPS [27] and CityPulse [28] exemplify this by optimizing resource management and employing distributed data analytics. FIWARE [30], FogFlow [33], and InterSCity [35] extend data processing across different nodes, with the latter specifically designed for large-scale applications. MiSCi showcases a multi-level structure for distributed processing [37], with OpenIoT also emphasizing cloud computing [38]. Similarly, S2NetM operates collaboratively [41], SEDIA integrates edge computing [42], and SGeoL employs tools such as Apache Kafka, all stressing the distributed nature of modern middleware [18]. As a final point, SmartCityWare further emphasizes this trend by distributing processes across cloud, fog, and IoT devices [45].

8) RESILIENCE AND FAULT-TOLERANCE

Resilience and fault-tolerance are significant non-functional requirements in smart city middleware to ensure consistent and robust operations. CityPulse incorporates a fault recovery component that provides estimated values for data streams when the quality diminishes or data is temporarily absent [28]. SmartCityWare demonstrates its resilience by automatically redirecting applications to available services when a sensor fails, ensuring uninterrupted service provision [45]. Similarly, the SmartSantander framework [46], through its testbed observation and management plane, emphasizes dynamic management and automated fault management. Upon detecting hardware failures, SmartSantander

reconfigures its system, excluding malfunctioning nodes from future tasks. Additionally, FIWARE is architected with resilience at its forefront, offering fault detection, recovery mechanisms, and additional provisions for fault tolerance [30].

9) USER INTERFACE

Middleware may come with interfaces for administrators, developers, or city officials to interact with. This includes dashboards, configuration panels, or developer tools. While the primary function of middleware is to act as a bridge between different software components, the user interfaces provided for its management, monitoring, and configuration are crucial for its efficient operation. CityPulse, for instance, offers users an immediate visual insight through its city dashboard, facilitating both holistic and intricate data views, while allowing real-time city monitoring [28]. OpenIoT places emphasis on user convenience, enabling the rapid development of IoT applications with minimal programming effort, thus indicating a user-friendly design [38]. The design of SEDIA emphasizes user interaction, as demonstrated by the air quality monitoring application [42]. Additionally, SGeoL includes a user-centric dashboard, suggestive of popular map-based applications, providing diverse features ranging from data visualization to geographic querying [18]. Snap4City provides varied interfaces, including city dashboards for decision-makers, smart city APIs, and data displays [47]. WeValue streamlines task creation for users through its TaskCreationGUI component, highlighting the importance of intuitive interfaces in middleware solutions [49].

10) DOCUMENTATION QUALITY

The success of a middleware solution, especially in complex smart city ecosystems, often depends on the quality of its accompanying documentation. Effective documentation ensures that administrators, developers, and other stakeholders understand how to use, deploy, configure, and optimize the middleware to align with specific urban challenges and objectives. For documentation to be of high quality, it should meet criteria such as clarity, completeness, accessibility, regular updates and feedback mechanisms. As a case in point, FIWARE offers comprehensive documentation for its components and APIs, emphasizing the platform's commitment to aiding developers in comprehending and utilizing the system effectively [30]. Platforms like FogFlow [33], InterSCity [35], and Snap4City [47] exhibit a strong dedication to thorough documentation on GitHub, emphasizing its crucial role in the effective adoption of smart city middleware solutions.

C. SUMMARY

In addressing the research question regarding the requirements of smart city middleware, our analysis, as outlined in Table 3 and Table 4, identifies a core set of both functional and non-functional features. These findings offer a comprehensive overview of what an ideal smart city middleware should encompass.

Among the functional requirements, data management, which includes Data Collection, Integration, Storage, Retrieval, and Data Processing, stands out as a foundational element in almost all middleware solutions. Service Management and Context Management further emphasize the importance of orchestrating services and understanding the context of data, ensuring that services remain agile and pertinent to evolving smart city contexts. Event and Alert Management, Resource Management, and Application Development, Deployment and Management are also integral, each being featured in a majority of the solutions. While Analytics and Reporting and IoT Management are present in a smaller subset, their importance within the smart city framework cannot be understated since they form the backbone for informed decision-making and seamless integration of a multitude of interconnected devices, respectively.

Regarding the non-functional requirements Scalability, Interoperability, Security and Privacy, and Distributed Processing are some of the predominant requirements across multiple middleware solutions, emphasizing their criticality in smart city contexts. Additionally, Configurability, Real-time Data Handling, Adaptability, Resilience and Fault-tolerance, User Interface, and Documentation Quality also emerge as significant features, though their presence varies across different middleware platforms.

Collectively, the functional and non-functional requirements detailed in Table 3 and Table 4 encapsulate the multifaceted roles smart city middleware plays in facilitating cities to be adaptive, efficient, and enabling cities to meet contemporary urban demands effectively. Notably, solutions like CityPulse [28] and FIWARE [30] exhibit a broad spectrum of these requirements, emphasizing their comprehensive approach to addressing the challenges of smart city ecosystems.

V. DESIGN AND PROGRAMMING PARADIGMS

A. ARCHITECTURAL STYLES

An architectural style of a system determines its core structure, the relationships of its components, and its design evolution principles. The following sections introduce the architectural styles observed in the reviewed middleware solutions. Table 5 provides a summary of the primary features of these architectural styles, listing both the benefits and challenges associated with each style.

1) SERVICE-ORIENTED ARCHITECTURE

SOA is a prevalent architectural style among the smart city middleware solutions examined [34], [36], [38], [40], [46], [48]. It facilitates the integration of various services, devices, and applications, regardless of the underlying technology, platform, or vendor, thereby promoting interoperability. Services can be dynamically discovered and accessed over a network, facilitating the integration of new services and

TABLE 3. Functional requirements for smart city middleware.

Smart City Middleware	Data Collection & Integration	Data Storage & Retrieval	Data Processing	Analytics & Reporting	IoT Management	Service Management	Context Management	Event & Alert Management	Resource Management	App Life Cycle Support
AMF-CPS	×	×	√	×	×	×	×	×	√	×
CityPulse	√	√	√	√	×	√	√	√	√	√
Civitas	×	×	√	×	×	√	×	√	×	√
FIWARE	√	√	√	√	√	√	√	√	√	√
FogFlow	√	√	√	√	√	√	√	√	√	√
GAMBAS	√	√	√	×	×	√	√	×	×	√
InterSCity	√	√	√	×	×	×	√	×	√	√
LinkSmart	√	√	√	√	√	√	√	√	×	√
MiSCi	×	×	√	×	×	√	√	√	√	×
OpenIoT	√	√	√	√	√	√	×	×	√	√
Rimware	×	×	×	×	√	√	×	×	×	×
S2NetM	√	√	√	×	×	√	√	×	×	×
SEDIA	√	√	√	×	√	×	×	√	×	√
SGeoL	√	√	√	√	√	√	√	√	√	√
SMArc	√	√	√	×	×	√	×	×	√	×
SmartCityWare	√	√	√	√	×	√	×	×	√	×
SmartSantander	√	√	√	×	√	×	×	√	√	×
Snap4City	√	√	×	√	×	×	×	√	×	√
SOUL	√	√	√	×	×	×	√	×	×	×
WeValue	√	√	√	×	×	√	×	×	×	√
Count	**16**	**16**	**18**	**8**	**8**	**14**	**10**	**10**	**11**	**12**

devices without significant changes to the existing system. Furthermore, services provide functionalities abstracted from the underlying implementation details, making it easier for developers to use and combine services without deep knowledge of their inner workings. Moreover, the modular design enhances maintainability and flexibility by enabling updates, replacements, or maintenance of individual services without impacting the entire system. On the other hand, the modularity and flexibility of SOA can result in increased complexity in the system's design, development, and management. Communication overhead between services over a network can lead to increased latency, a concern for time-sensitive applications. Additionally, exposing services over a network broadens the attack surface, necessitating careful design and implementation of security measures.

2) DISTRIBUTED ARCHITECTURE

Distributed architecture can enhance the functionality of middleware solutions by enabling peer-to-peer interaction, decentralized control, and fault tolerance as demonstrated by middleware solutions like SGeoL [18], S2NetM [41] and SmartCityWare [45]. It can lead to better performance as the load is distributed among all the nodes or components, and there is no single point of failure [33], [42]. However, increased communication overhead can be a concern. The architecture allows for horizontal scalability, which is particularly important for smart city applications that may involve a large number of nodes or components [28]. The decentralized nature of the architecture can positively impact the robustness and flexibility of the system. However, it also comes with challenges related to complexity, consistency, interoperability and security.

3) MICROSERVICES

Microservices model is an evolution of the SOA model that is characterized by the development of small, independent, and deployable services, with each microservice running in its own process and communicating with others using lightweight mechanisms, often an HTTP API or a messaging queue [90]. While SOA and microservices share many principles, the microservices model emphasizes fine-grained services, independent deployment, and decentralized governance, setting it apart as a distinct architectural paradigm. Among the strengths of this architectural style, resource optimization and system performance are notably enhanced by the ability to scale individual services independently. This granular level of scalability not only optimizes resource utilization but also contributes to a more resilient system architecture, where the deployment and updates of services can occur independently of one another. Yet, the increased

TABLE 4. Non-functional requirements for smart city middleware.

Smart City Middleware	Scalability	Interoperability	Security & Privacy	Adaptability	Configurability	Real-time Data Handling	Distributed Processing	Resilience & Fault-tolerance	User Interface	Documentation Quality
AMF-CPS	×	√	×	×	×	×	√	×	×	×
CityPulse	√	√	×	√	√	√	√	√	√	×
Civitas	×	×	√	√	×	×	×	×	×	×
FIWARE	√	√	√	√	√	√	√	√	×	√
FogFlow	√	√	×	×	√	√	√	×	×	√
GAMBAS	×	√	√	√	×	×	√	×	√	×
InterSCity	√	√	×	√	√	√	√	×	×	√
LinkSmart	×	√	×	√	√	√	√	×	√	×
MiSCi	√	√	×	√	√	×	√	×	×	×
OpenIoT	×	√	√	√	√	√	√	×	√	×
Rimware	×	√	√	√	×	×	×	×	×	×
S2NetM	×	√	√	×	×	×	√	×	√	×
SEDIA	√	√	√	×	×	√	√	×	√	×
SGeoL	√	√	√	×	×	√	√	×	√	×
SMArc	√	√	√	√	×	×	×	×	×	×
SmartCityWare	√	×	√	√	√	√	√	√	×	×
SmartSantander	×	√	√	×	√	×	×	√	√	×
Snap4City	√	√	√	×	×	×	×	×	√	√
SOUL	√	×	×	×	×	√	√	×	×	×
WeValue	×	×	√	×	×	×	√	×	√	×
Count	**11**	**16**	**13**	**11**	**9**	**9**	**15**	**4**	**10**	**4**

complexity of managing multiple services can be a significant trade-off. Middleware solutions like FIWARE [30], InterSCity [35] and Snap4City [47] have adopted this architectural style, integrating tools that simplify the management, monitoring, and orchestration of the developed microservices.

4) AGENT-BASED ARCHITECTURE

Agent-Based Architecture (ABA) is characterized by the use of intelligent agents, which are software entities that can make decisions autonomously, interact with each other, and adapt to changes in the environment [91]. ABA inherently enhances functionality by creating a system capable of autonomously adapting to operational changes or new requirements. Despite the potential for increased latency and overhead due to agent communication and coordination, the decentralized structure of ABA distributes the load across multiple agents, potentially enhancing overall performance. This decentralized nature also lends itself to horizontal scalability, as adding new agents does not drastically alter the existing system. However, this approach also brings increased complexity and security concerns, requiring meticulous attention to non-functional requirements like security, maintainability, and robustness. AMF-CPS is an example of middleware that follows the ABA [27]. It maintains an agent repository to generate agents, ensuring reliable service response and request handling for application layer users. Another instance of middleware adopting the ABA is WeValue [49], where a 'Value Agent' actively represents a stakeholder. This agent participates in task matching and orchestration processes, working to identify, negotiate, and approve appropriate task compositions. The MiSCi middleware is also aligned with the ABA approach [37].

5) BLOCKCHAIN-BASED ARCHITECTURE

Blockchain-Based Architecture (*BBA*) is a decentralized system [84]. Instead of relying on a single central authority or location for data storage, the ledger in BBA is distributed across a network of participating nodes. BBA has the potential to enhance the functionality of middleware solutions in smart city applications by facilitating transparent, secure, and automated transactions through mechanisms like smart contracts [26]. This fosters trust among different smart city stakeholders. On the other hand, the significant computational power needed for Blockchain transactions can affect performance, especially in applications that demand real-time responses. Additionally, scalability can be a concern, as the computational demands typically increase with the Blockchain's size. The WeValue middleware examined in this study is a case of BBA [49]. WeValue operates based on a Blockchain-based smart contract that formally specifies the

TABLE 5. Comparison of the different architectural styles in middleware solutions.

Architectural Style	Benefits	Challenges
Service-Oriented	Integration, Interoperability, Flexibility, Maintainability, Platform independence, Reusability, Scalability	Complexity, Performance, Security, Service discovery, Service versioning, Transaction management
Distributed	Concurrency, Decentralization, Fault tolerance, Flexibility, Load balancing, Robustness, Scalability	Communication overhead, Complexity, Consistency, Interoperability, Maintainability, Security
Micro-services	Development productivity, Fault isolation, Fine-grained services, Independent deployment, Lightweight communication, Scalability	Data consistency, Inter-service communication, Monitoring and logging, Service coordination
Agent-Based	Adaptability, Autonomy, Decentralization, Flexibility, Interactive behavior, Robustness, Scalability	Communication overhead, Complexity, Coordination, Resource consumption, Security, Unpredictability
Blockchain-Based	Decentralization, Immutability, Reduction of intermediaries, Security, Smart contracts, Transparency, Trust	Complexity, Energy consumption, Integration with existing systems, Performance, Regulatory uncertainty, Scalability
Hybrid	Adaptability, Flexibility, Performance, Versatility	Increased complexity, Potential compatibility issues, Requires expertise

agreement for a collaborative activity, encoding contractual obligations, rewards, and penalties.

6) HYBRID ARCHITECTURE

A hybrid architecture offers significant flexibility as it permits the integration of different architectural styles, thereby leveraging the benefits of each [92]. Such a design can be customized to address the unique needs and specifications of a smart city middleware solution. However, the design and implementation of a hybrid architecture can be more intricate. It necessitates a profound understanding of the distinct styles being integrated and their potential interactions. In this study, two cases were identified that adopt a hybrid architecture. The MiSCi middleware integrates both SOA and ABA [37], whereas WeValue integrates Agent-based approaches with BBA-centric contracts [49]. Another mixed instance is FIWARE [30], which melds SOA with microservices, though these operate under the same guiding principle.

B. PROGRAMMING PARADIGMS

A middleware aims to provide a suite of programming abstractions that facilitate software development, especially when there is a need to integrate and establish communication among heterogeneous components. The subsequent sections delve into the programming models discerned in the reviewed middleware solutions.

1) SERVICE-ORIENTED PROGRAMMING

Service-oriented programming often arises naturally from adopting a service-oriented architecture, though it can also manifest other architectural styles such as ABA. In this programming paradigm, applications are structured as a collection of loosely coupled, independently deployable services, exemplifying modularity. A middleware solution can furnish developers with a suite of services to aid in application development. For instance, AMF-CPS provides services facilitated by a collection of specialized agents, offering specific functionalities such as resource allocation and scheduling [27]. FIWARE provides a set of services (or microservices) that developers can use to manage data, devices, security, and other aspects of their applications [30]. Similarly, SmartCityWare provides core services (e.g., invocation, location-based, fault-tolerant, and security services) to effectively build smart city applications [45].

2) DATA-CENTRIC PROGRAMMING

Data-centric programming offers a high-level abstraction centered on data and its flow through the application, rather than the control flow [93]. This abstraction encapsulates several programming models: processing data as it becomes available, handling continuous streams of data, and modeling applications as directed graphs that depict the flow of data between operations. Key facets of this abstraction include:

- Data Stream Programming: Focuses on real-time or near-real-time processing of continuous data streams.
- Data-Driven Programming: Emphasizes processing data as it becomes available, without necessarily involving real-time processing.
- Dataflow Programming: Concentrates on modeling the application as a directed graph of the data flow between operations, promoting modularity, parallelism, and facilitating understanding of the data flow.

These aspects offer a unified perspective adaptable to a variety of specific requirements and use cases. For example, the CityPulse platform provides services for collecting, processing, and analyzing data streams in real-time, facilitating the development of applications focused on real-time data stream processing [28]. Similarly, the SOUL middleware adopts a data stream reasoning model, which was implemented using real-time big data processing technology [48]. On the other hand, Snap4City adopts a data-driven programming model for creating IoT-based applications [47]. An enhanced form of the dataflow programming model is proposed by FogFlow [33]. Specifically, it introduces special operators based on NGSI to augment the traditional dataflow programming model, thereby enabling more efficient fog computing.

3) EVENT-DRIVEN PROGRAMMING

The event-driven programming model is based on the concept of event listeners and event handlers [94]. The middleware can provide a set of events (e.g., data received, device status changed, etc.) and allow developers to define handlers for those events. This approach is particularly suitable for applications that need to respond to real-time changes in the environment. Additionally, the suitability of this paradigm for smart city applications is emphasized by its inherent capabilities, including efficient resource utilization, enhanced modularity and maintainability, seamless integration of heterogeneous systems, and support for an enhanced user interaction experience. As an anticipated consequence, many of the middleware solutions examined in this study, including Civitas [29], LinkSmart [36], SEDIA [42], and SmartSantander [46], have embraced this model.

4) MASHUP

The Mashup model involves combining data, presentation, or functionality from two or more sources to create new services [95]. The middleware can provide APIs and tools for accessing and integrating data and services from various sources, enabling developers to create mashup applications that combine and reuse data and services from multiple sources. The SGeoL platform exemplifies a mashup application development approach, evident from its integrative and composite nature [18]. It facilitates the combination of diverse functionalities like layer creation, geographic querying, and data visualization into a single, unified interface, making it characteristic of web mashups. Moreover, with the provision of high-level RESTful APIs, developers can seamlessly combine content from multiple sources, thus creating integrated applications. OpenIoT also supports a mashup approach, offering a zero-programming environment through a set of visual tools that streamline the development process [38]. The WeValue platform seamlessly merges distinct client-side and backend components, demonstrating the integration characteristic of a mashup development approach [49].

C. SUMMARY

Table 6 offers a comprehensive overview of the diverse architectural styles and programming paradigms embraced by the surveyed smart city middleware solutions. Recognizing that architectural classifications can vary based on distinct perspectives, it becomes evident that no singular classification scheme is universally applicable. Given this context, this study emphasizes the most salient architectural characteristics to categorize the solutions.

While Distributed and Service-Oriented architectures appear to be prevalent choices, Agent-Based and Microservices architectures are also notably represented. In contrast, Blockchain-Based architecture remains a more exploratory approach but holds promising potential for future platforms in the smart city domain. Hybrid architectural styles, such as the combination of Service-Oriented with Microservices (as seen in FIWARE) or Blockchain-Based with Agent-Based (as in WeValue), suggest a move towards leveraging the strengths of multiple paradigms for enhanced functionality and flexibility.

In terms of programming models, there is a clear trend towards Service-Oriented approaches, with many middleware solutions also incorporating Event-Driven and Data-Centric models. This mix indicates a need for versatile middleware capable of handling diverse data types and flows in real-time smart city scenarios. A few platforms, like SGeoL and WeValue, also adopt the Mashup model, reflecting the perspective that integrating diverse data sources and services seamlessly is an efficient approach to support the development of smart city applications.

Table 6 highlights also the expansive application range covered by these middleware solutions. They span a variety of sectors, from simulations, smart energy, and smart mobility to environmental monitoring, smart government, and intelligent traffic management. Notably, middleware platforms like FIWARE, OpenIoT and SmartSantander exhibit a broader adaptability, catering to an extensive array of smart city sectors.

VI. CHALLENGES AND OPEN ISSUES

In this section, the primary challenges associated with using middleware solutions in building smart city applications are discussed, as identified by the reviewed literature. Each highlighted challenge represents an open issue that warrants further investigation and holds significant promise for future research.

A. INTEROPERABILITY AND STANDARDIZATION

Given the diverse array of systems, protocols, and standards in smart cities, achieving seamless integration remains a significant challenge [96]. Among the smart city middleware solutions examined, SGeoL [18], SNetM [41], and SMArc [44] stand out as characteristic examples that attempt to leverage Semantic Web technologies and Ontologies to tackle the problem of heterogeneity; however, achieving a universal semantic understanding across diverse systems and maintaining up-to-date ontologies in rapidly evolving urban environments pose persistent obstacles [19].

On the other hand, many technology providers offer proprietary solutions with their own set of standards and protocols. Middleware must bridge the gap between these proprietary solutions and open standards to achieve a cohesive smart city ecosystem. Therefore, while solutions like FIWARE [30] and OpenIoT [38] aim for open standardization, they must also consider integrations with vendor-specific solutions.

Devices in smart cities might communicate using a plethora of protocols like MQTT, CoAP, HTTP/HTTPs, Zigbee, and LoRaWAN [51]. Middleware solutions face the challenge to understand and translate between these protocols, ensuring seamless data flow. For instance, the SEDIA platform proposes the use of Protocol Translation Gateways to convert the data into a format that the middleware can

TABLE 6. Summary of design and programming paradigms in smart city middleware.

Middleware	Architectural Style	Programming Model	Main Applications
AMF-CPS	Agent-Based	Service-Oriented	Simulation using the OPNET simulator
CityPulse	Distributed	Service-Oriented; Event-Driven; Data-Centric	Smart travel planner; smart energy; smart safety
Civitas	Distributed	Service-Oriented; Event-Driven	Smart car movement prediction and tracking
FIWARE	Service-Oriented; Microservices	Service-Oriented; Event-Driven; Data-Centric	Extensively used in smart city projects and various application sectors
FogFlow	Distributed	Data-Centric; Service-Oriented	Anomaly detection of energy consumption; Video surveillance in stadiums
GAMBAS	Service-Oriented	Service-Oriented	Linked weather
InterSCity	Microservices	Service-Oriented	Smart parking; Environmental monitoring
LinkSmart	Service-Oriented	Service-Oriented; Event-Driven	Smart energy; Smart building; Smart health
MiSCi	Service-Oriented; Agent-Based	Service-Oriented	Simulation of smart city scenarios: Emergency mobility system activation for subway failures
OpenIoT	Service-Oriented	Mashup	Smart mobility and transportation; Smart waste; Environmental monitoring
Rimware	Service-Oriented	Service-Oriented	Personal training, heart rate monitor and smart lighting
S2NetM	Distributed	Service-Oriented	Smart mobility; Smart home; Smart health
SEDIA	Distributed	Service-Oriented; Event-Driven	Urban air quality monitoring and navigation
SGeoL	Distributed	Mashup	Smart government; Urban planning management
SMArc	Distributed	Service-Oriented	Smart Grid
SmartCityWare	Distributed	Service-Oriented	Intelligent traffic light control system; Smart buildings collaborative data analytics
SmartSantander	Service-Oriented	Service-Oriented; Event-Driven	Extensively used in various smart city application domains
Snap4City	Microservices	Data-Centric; Mashup	Smart mobility/transportation
SOUL	Service-Oriented	Data-Centric	Smart city applications with stream reasoning, such as fire accident management
WeValue	Blockchain-Based; Agent-Based	Mashup	Co-creation, collaboration, and citizen engagement activities

understand [42]. However, implementing and maintaining these gateways introduces complexities, can impact real-time data processing, and requires constant updates to adapt to emerging communication standards.

Furthermore, many cities have older, legacy infrastructure systems that weren't originally designed for modern interconnected environments [99]. Integrating these systems with newer technologies without compromising their operations is an additional complex challenge.

B. SCALABILITY

Platforms such as FogFlow [33] and InterSCity [35] are designed with scalability in mind, but there are always trade-offs between scalability and other attributes. For instance, performance is one such attribute that can be compromised as systems scale [97]. As they grow, systems may experience increased latency or reduced throughput. In such scenarios, managing thousands of simultaneous device connections could lead to slower response times, which in turn could affect real-time decision-making processes crucial for specific smart city operations.

Resource management is another relevant attribute [98]. As more devices are added, efficient allocation and utilization of resources, including computational power, memory, and bandwidth, become more complex and crucial. More devices mean more network traffic. Middleware solutions must optimize data transmission to prevent network congestion and ensure timely delivery of critical information. In this context, SEDIA employs Adaptive Data Rate (ADR) and task scheduling to ensure efficient communication under extreme conditions [42]. However, consistently achieving this balance, especially during peak data traffic or in situations of sudden influx of data, remains a challenging endeavor for most middleware platforms. Even with workload management strategies, there is a constant need for real-time adjustments and monitoring to maintain efficient communication throughout the smart city infrastructure [100].

While cloud infrastructure is utilized by the vast majority of the examined middleware solutions to achieve scalability, it also entails cost implications, as suggested in SmartCityWare [45]. As usage increases, so do operational costs, which can be a limiting factor for many city administrations.

Building on these scalability considerations, Dew/Mist Computing emerges as a further evolution in the domain [101], [102]. It focuses on leveraging ubiquitous mobile devices for distributed processing, thereby expanding the scope and capabilities of smart city middleware solutions. This self-sustaining computing infrastructure taps into the collective power of distributed devices, accommodating both real-time and batch processing. This approach offers a

cost-effective solution to the data management challenges of smart cities, potentially mitigating issues related to network congestion and the high operational costs associated with cloud infrastructure, as highlighted in SmartCityWare.

C. SECURITY AND PRIVACY

The challenge of ensuring security and privacy in the realm of smart city middleware is multifaceted, involving both technical and socio-organizational dimensions [20]. Even as middleware solutions like Rimware [40] and Snap4City [47] may employ state-of-the-art encryption protocols to safeguard data during transit, the increasing sophistication of cyber-attacks requires constant awareness and adaptation. As seen with middleware solutions like S2NetM [41], there is an ongoing need to ensure encryption methodologies can withstand novel decryption techniques and brute-force attacks.

Protecting individual identities and sensitive information necessitates the anonymization or aggregation of data before its use in broader applications or analytics [103]. Middleware solutions such as GAMBAS [34] and SmartCityWare [45] emphasize the need for advanced techniques to ensure that individual data points in aggregated datasets cannot be reverse-engineered.

In urban settings, navigating data privacy challenges requires middleware solutions to balance technical data protection with ethical considerations and adhere to regulations like the General Data Protection Regulation (GDPR) in Europe [104]. Furthermore, as civic bodies and stakeholders push for more transparent governance, there is an increasing demand for middleware that can seamlessly integrate privacy-preserving measures without compromising the functionality or user experience of smart city applications [105].

D. CONTEXT MANAGEMENT

Context acquisition, modeling, reasoning, and distribution in the realm of a smart city present significant challenges for middleware solutions due to the size and complexity of data sources and the need for real-time, adaptive, and accurate interpretations [106]. S2NetM [41], for example, examines the challenges associated when converting raw context data into actionable knowledge. This requires exploring interpretation mechanisms like rule-based reasoning, ontology-based approaches or more advanced machine learning techniques to specify suitable interpretation methods and reasoning/aggregation mechanisms aligned with the context model's capabilities [106]. On the other hand, FogFlow [33], explores various methods to ensure the distribution of context information, including proximity-based and node-grouping approaches. Challenges related to context distribution include the stability and coherence of the data as it is distributed across nodes, such as determining the information's validity over time and handling potentially conflicting data from multiple observing nodes [107].

SGeoL advocates that a unified city model is required to facilitate the representation, manipulation, comprehension, and integration of heterogeneous context data [18]. Such a model faces challenges in scalability, managing temporal and spatial dynamics, ensuring data integrity, promoting semantic understanding, and maintaining privacy and security [17], [97].

E. BIG DATA MANAGEMENT AND STORAGE

With the vast amounts of data generated in smart cities, efficient data storage, retrieval, and processing become critical [63]. Platforms such as CityPulse [28] and SmartCityWare [45] have attempted to provide effective data management strategies. However, there are several associated challenges that platforms need to address [108]. For example, managing data quality is pivotal for effective big data management and storage, given the diverse sources of data and the immediacy of data-driven decisions impacting citizens [38], [109]. Addressing challenges like noise, inconsistencies, timeliness, and scalability in data, while ensuring its verification and traceability, is essential to maintain the efficacy of smart city solutions and retain citizens' trust.

Moreover, the increasing volume and complexity of geospatial data from diverse sources pose significant challenges in storage, real-time processing, integration, and visualization [110]. Middleware solutions like SGeoL [18] and SEDIA [42], which have demonstrated the use of geospatial data in smart city applications, underline the need to address these complexities to ensure the efficient, secure, and accurate delivery of critical geographic information across various application domains.

Designing smart city services from urban big data involves intricate challenges, from managing diverse data sources and ensuring real-time processing to adopting user-centric designs for efficient information delivery [111]. Key considerations include data standardization, scalable processing, structured storage, visualization techniques, multi-platform accessibility, and the maintenance of security and privacy, as highlighted by the CityPulse [28] and LinkSmart [36] cases.

F. RESILIENCE AND FAULT TOLERANCE

To enhance fault tolerance, middleware solutions often adopt a distributed architecture, leaning heavily on cloud infrastructures to provide fault-tolerance services as in the case of SmartCityWare [45] and SmartSantander [46]. Yet, merely having a distributed framework is insufficient. Methods and mechanisms are essential to bolster resilience and fault-tolerance [112]. For example, reactive methods are required such as: data replication to ensure data availability even when certain nodes fail; load balancing to avoid overburdening particular nodes; task resubmission to guarantee the completion of critical tasks; or checkpointing to allow the system to return to a safe state post-failure. Furthermore, proactive methods can be employed to introduce more foresight

such as: self-healing capabilities to automatically detect and fix faults [113]; preemptive migration to shift tasks from potentially failing nodes [114]; or fault prediction through using patterns to anticipate and prevent future issues [115]. Going a step further, resilient methods employ machine learning, particularly reinforcement learning, to interact with the environment and dynamically adapt fault-handling strategies [116]. This learning-oriented approach can enhance the middleware's ability to cope with unexpected challenges efficiently.

G. QUALITY OF SERVICE

Quality of Service (QoS) stands out as a critical parameter in the design and implementation of smart city services [117]. QoS encompasses various parameters, including real-time services, SLAs (Service Level Agreements), and other metrics to ensure that services meet the desired performance and reliability standards. For example, middleware solutions such as CityPulse [28] and SOUL [48] have emphasized the processing of data in real-time to support applications like traffic management or emergency response. However, real-time services require middleware to process, analyze, and deliver data with minimal delay. This is particularly challenging given the diverse and massive amount of data that smart cities generate. Furthermore, to maintain QoS middleware must adapt to dynamic resource demands, especially during data surges, and address network variabilities by potentially rerouting data or using backup channels [118].

Ensuring SLAs poses significant challenges [25]. SLAs in smart cities necessitate middleware adaptation capabilities to meet diverse service requirements, such as prioritizing response time for emergency services or ensuring data accuracy for public transport. This entails attentive monitoring of performance, with deviations triggering alerts and requiring corrective measures. Using Blockchain technology, middleware solutions like WeValue [49], which support smart contracts, can provide transparent tracking of SLA adherence, potentially reducing costly penalties for violations.

H. ENERGY EFFICIENCY

Particularly for IoT and mobile devices, energy consumption can be a concern. As indicated in the OpenIoT framework [38], it is important to develop energy-efficient data collection, transmission, and processing strategies. Middleware solutions need to determine optimal data transmission frequencies or use adaptive communication strategies as proposed in the SEDIA platform [42] and to strike a balance between speed and energy, possibly by employing lightweight algorithms for initial data processing [119].

Dynamic energy management in smart cities involves intricate challenges, including monitoring real-time energy usage of diverse devices and adapting to the unpredictable urban environment. Strategies like energy-aware task scheduling, energy-efficient protocols, offloading to the cloud, and edge computing each bring unique benefits and challenges, from ensuring quality of service and minimizing energy overheads to managing computational demands and transmission costs [120].

I. EVOLUTION AND MAINTENANCE

Middleware solutions need to support a wide range of applications, from environmental monitoring to smart mobility and transportation as indicated by solutions like FIWARE [30], OpenIoT [38] and SmartSantander [46]. Tailoring the middleware to be general enough for various applications yet specific enough for optimal performance is a challenge requiring meticulous design and continuous evolution and maintenance. The dynamic and heterogeneous nature of urban data sources, combined with the diverse functional requirements of each application, necessitates the development of modular, scalable, and interoperable middleware components [21], [120]. Furthermore, middleware solutions in smart cities should be able to be updated and expanded without disrupting existing services, necessitating seamless rollouts of updates and patches [17].

VII. CONCEPTUAL FRAMEWORK

Based on the responses to the survey research questions and the analysis of the 20 middleware solutions, a conceptual framework for smart city middleware has been framed, as depicted in Figure 2. This framework aims to define the essential features required for the development of an effective middleware platform, facilitating the creation of integrated and scalable smart city applications.

The formulated conceptual framework presents a holistic view, integrating both the functional and non-functional requirements pivotal for the effective operation of middleware solutions. Alongside these, the framework sheds light on the relevant challenges, detailing various pertinent topics that often emerge as obstacles in middleware design and deployment. The subsequent sections delve into an exploration of these multifaceted topics. The aim is to offer a clearer understanding of the current landscape and the inherent challenges that developers and city planners must handle. Furthermore, given that technological advancement is a continuous process, this discussion offers insights into potential future developments, suggesting potential paths for the continued evolution of middleware platforms.

A. FUNCTIONAL TOPICS

1) DATA COLLECTION AND INTEGRATION

Managing data heterogeneity necessitates normalization protocols, the adoption of industry-standard models, and adaptable middleware extensions. To achieve third-party interoperability, a versatile API framework, comprehensive documentation, and compatibility with open data platforms are required. For real-time data handling, middleware should focus on low-latency processes, ensure scalability, manage data surges, and integrate automated validation, machine

learning for anomaly detection, and alert mechanisms to assure data quality.

2) DATA STORAGE AND RETRIEVAL

Adaptive storage solutions are essential, with evaluations of relational to graph-based databases, and hybrid systems blending various advantages. Maintaining data quality necessitates automated cleaning tools, real-time verification, traceability, collaboration with data providers, and consistent monitoring. Enhancements in data retrieval encompass efficient indexing, in-memory processing, semantic storage, AI-driven pattern recognition, and specialized geospatial data management strategies including visualization and security protocols.

3) DATA PROCESSING

Efficient data management requires dynamic algorithms for filtering and selection, middleware capabilities for data transformation and normalization using AI, and robust mechanisms for data fusion without compromising integrity. Additionally, middleware should offer versatile data processing options, adaptability to changing city needs, and leverage advanced analytics for optimized results.

4) ANALYTICS AND REPORTING

Dynamic visualization methods should cater to diverse users, offering real-time urban insights and deep data exploration. Middleware must offer customizable reporting, combining templated and user-defined formats, and incorporate predictive analytics with automated alerts for anomalies. It is vital that middleware embeds AI for robust data analysis, providing on-the-spot recommendations and understanding urban situations comprehensively. Collaborative analytics in middleware should foster multi-entity collaboration, harness distributed computing, and emphasize data security. Lastly, middleware should support multiple analytical platforms, facilitate integration with new tools, and support open standards for effortless third-party interfacing.

5) IoT MANAGEMENT

Middleware for device onboarding should standardize IoT integration, handle diverse device types, and ensure compatibility with both old and new technologies. It must also offer remote adaptability, support timely software updates, and empower devices to self-adjust based on centralized guidance. Continuous device monitoring, full lifecycle management, and predictive maintenance algorithms are essential in middleware to preemptively tackle device issues. Additionally, middleware should embrace various IoT communication standards, facilitate smooth channel transitions, and maximize bandwidth efficiency in connectivity-constrained settings.

6) SERVICE MANAGEMENT

Middleware in smart cities should autonomously detect new services and devices, use semantic understanding for contextual depth, and ensure modular application design through dynamic service composition. Service orchestration mandates protocols for diverse device communication, scalability for growing city networks, and feedback-driven real-time performance enhancement. Rapid service deployment, holistic monitoring, and oversight of interconnected systems are essential for meeting urban demands. Lifecycle management must cover a service's full duration, from start to end, and use contract-based collaboration methods. Lastly, involving all stakeholders, from providers to citizens, is crucial in the service management process.

7) CONTEXT MANAGEMENT

Turning raw data into actionable insights necessitates processing and quality assurance, the creation of context-discerning algorithms, the right blend of interpretative methods, continuous method refinement, and user-friendly visualization tools for decision-makers. Distributing context information efficiently calls for advanced data transfer techniques, maintaining data integrity, systems designed for fail-safety, prioritized data sharing, and adaptability to network conditions. In collaboration with experts, a comprehensive city model should encapsulate various contextual elements, be scalable, handle temporal and spatial details, provide multi-layered data interpretation, and strictly adhere to privacy and security norms.

8) EVENT AND ALERT MANAGEMENT

In smart city middleware, efficient algorithms combined with AI and machine learning are essential for real-time event detection, focusing on scalability and accuracy. Precise communication for these detections involves middleware optimization, geopositional tagging, and assured timestamping. Event identification is further refined with semantic tools, ontologies, and semantic web technologies to deepen context comprehension. Managing these events optimally demands distributed middleware solutions like the topic-based model for resilience and adaptability. A comprehensive alert system is essential, prioritizing personalized notifications, elevating significant irregularities, and featuring user-friendly interfaces, potentially augmented with visualization tools, to ensure rapid decision-making.

9) RESOURCE MANAGEMENT

Middleware design should prioritize adaptive allocation algorithms and demand a balance between centralized and decentralized management for efficiency and reliability. Managing diverse computational, storage, and physical resources necessitates tailored strategies and an exploration of hybrid cloud-edge computing for task allocation. Stream processing presents unique challenges, requiring insights into real-time tasks, understanding the benefits of edge-based processing, and flexibility in choosing processing locations. The integration of logical and physical resource management calls for creating logical resource abstractions. Additionally,

middleware architectures must be versatile enough to support both resource types while ensuring their synchronization.

10) APP LIFE CYCLE SUPPORT

Smart city application development requires several foundational pillars. A developer-friendly environment is central, necessitating comprehensive tools, rapid prototyping methods, and adaptable modular designs. Life-cycle management requires robust version control, proactive health monitoring, and efficient recovery mechanisms to minimize service disruptions. Decentralized application development requires collaborative platforms, with an acute awareness of security challenges, especially in open-source environments, and an emphasis on community feedback. User-centric designs demand thorough engagement with city inhabitants, standardized data handling, swift data processing, and user-intuitive interfaces.

B. NON-FUNCTIONAL TOPICS

1) SCALABILITY

Balancing scalability and performance centers on employing adaptive architectures, real-time monitoring, and discerning the intricacies of centralized versus decentralized setups. This scalability is deeply intertwined with resource management, which requires advanced allocation algorithms, anticipatory actions against demand surges, and dynamic bandwidth solutions to accommodate a growing number of devices. Similarly, as scalability grows, so does the challenge of real-time workload management, necessitating middleware that evenly distributes tasks, has failover provisions for unexpected data inflows, and harnesses predictive analytics. Furthermore, city administrations should assess the economic implications of such scalability on cloud platforms, embracing hybrid solutions and cultivating vendor relationships for cost-effective approaches.

2) INTEROPERABILITY

Achieving interoperability demands adaptive middleware and cross-domain ontologies for robust semantic comprehension, with continuous automation for up-to-date ontologies. To bridge the gap between proprietary systems and open standards, versatile middleware is pivotal, alongside fostering collaborations to enhance transparency. Efficient data interchange centers on thorough evaluation of Protocol Translation Gateways, broad protocol support, and rigorous scenario testing. For legacy systems, a tailored middleware approach, modern retrofitting, and careful technology integration are crucial to preserve security and functionality.

3) SECURITY AND PRIVACY

Middleware design must prioritize security and privacy, integrating early expert collaboration, user consent, and consistent reassessment. Adopting advanced data anonymization involves differential privacy, immediate data anonymization, and defenses against de-anonymization. Middleware should offer transparent data processing views without compromising privacy, backed by role-based access and public awareness efforts. Staying compliant requires integrating regional data rules, transparency for audits, and adaptability to legal shifts. Engagement with the public and experts, alongside granular user controls and regular policy reviews, ensures ethical data handling.

4) ADAPTABILITY

To foster a responsive and adaptive smart city framework, middleware solutions must rapidly adapt to urban changes and citizens' immediate behaviors. This requires a foundation on modular and open architectures, ensuring flexibility, easy third-party integrations, and the addition of new features without service disruptions. Behavioral analytics are crucial for understanding and enhancing user experience while maintaining privacy. Data handling demands tools that cater to diverse formats and guarantee consistent quality across varied sources. Scalable gateway adapters augment interface adaptability, optimizing latency for real-time functions. Furthermore, staying updated necessitates strategies for ontological evolution, complemented by the latest in AI and machine learning.

5) CONFIGURABILITY

Middleware platforms should emphasize modular design with selective component deployment to achieve enhanced configurability. Developers benefit from these configurable platforms that offer intuitive interfaces and automation for application-specific settings. A user-focused design approach is key, integrating streamlined development tools and adapting configurations based on user feedback. To maintain top-tier performance, continuous configuration evaluations, real-time analytics, and expert collaboration are essential, ensuring user-centric metrics are met.

6) REAL-TIME DATA HANDLING

To support real-time data handling in middleware design, it is crucial to prioritize instant metric monitoring, flexible algorithms, and a harmonious trade-off between speed and energy efficiency. Addressing fluctuating data publication rates, especially during high traffic, demands the integration of buffering and caching techniques for continuous real-time processing. Achieving instantaneous semantic interpretation requires the fusion of state-of-the-art semantic engines, optimized labeling processes, and AI-driven contextual analysis. Finally, real-time complex event analysis mandates the adoption of advanced processing tools, a careful balance of analytical depth and responsiveness, and interdisciplinary collaboration for refining event detection parameters.

7) DISTRIBUTED PROCESSING

Middleware in smart cities must handle edge, fog, and cloud hierarchies for efficient data processing. This requires dynamic task allocation across computing nodes with a strong emphasis on secure communications to ensure data integrity.

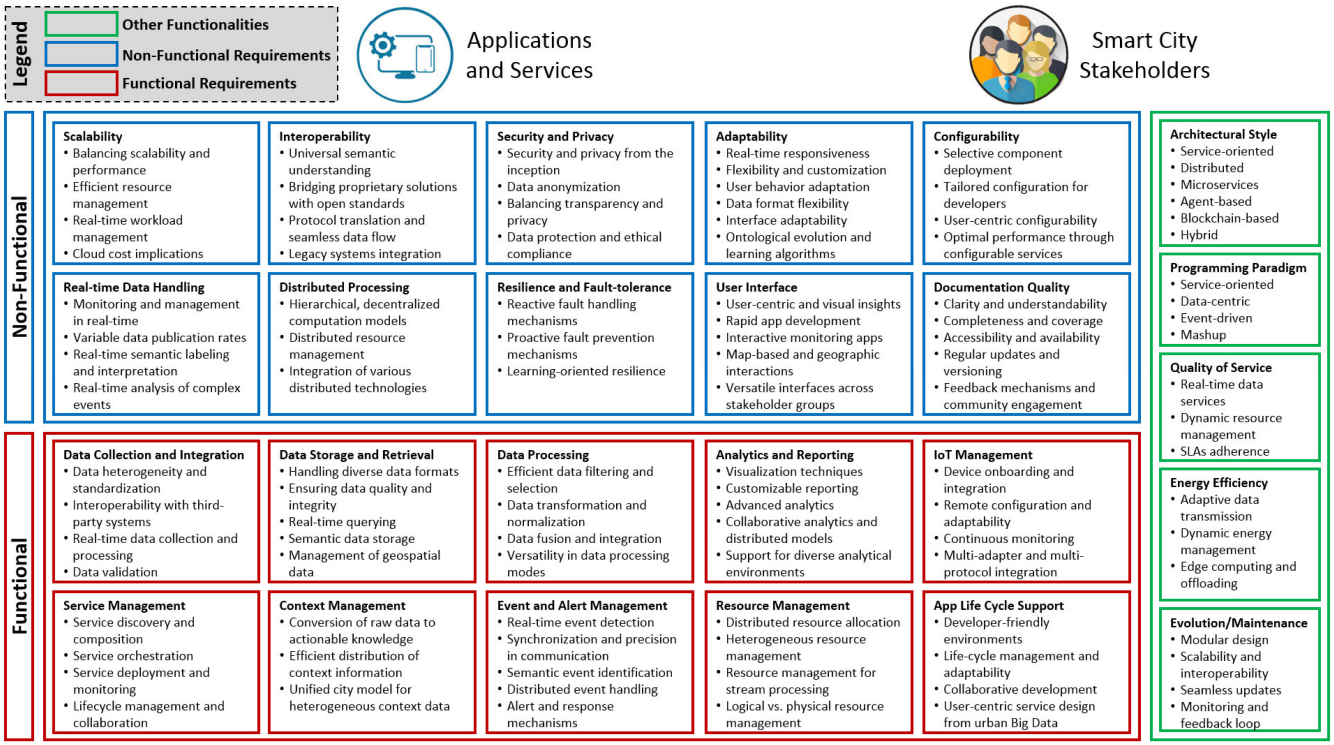

FIGURE 2. **Conceptual framework outlining functionalities prevalent in existing solutions complemented by essential topics that should be considered for the optimal development of future smart city middleware.**

Optimal processing pivots on resource allocation that prevents bottlenecks, focuses on energy efficiency, and has robust failover mechanisms. Furthermore, versatility is key, with the integration of technologies like Apache Kafka and a continuous evaluation of performance, ensuring interoperability across various platforms and standards.

8) RESILIENCE AND FAULT-TOLERANCE

In the smart city domain, middleware must prioritize reactive fault handling through efficient data replication, dynamic load balancing, task resubmission protocols, and rapid recovery using checkpoint mechanisms. Proactively, middleware should feature self-healing capabilities, predictive task migrations, and fault prediction algorithms, bolstered by collaborations with predictive analytics experts. For enhanced resilience, the integration of reinforcement learning modules, paired with domain expert collaborations, periodic model updates, and feedback loops, is crucial to ensure middleware continually adapts to challenges and real-world feedback.

9) USER INTERFACE

In the context of smart cities, middleware is crucial for enhancing user experience and system functionality. It should merge user-centric design, tailored for city administrators, with dynamic visualization and real-time monitoring to swiftly address anomalies. While designed for simplicity, middleware should mask technical complexities for novices yet retain expert functionalities. The system ought to foster interactive monitoring, adapt to user behaviors, and provide timely alerts. With geospatial integration and diverse user-centric interfaces, feedback mechanisms should further enrich its multifaceted approach.

10) DOCUMENTATION QUALITY

Middleware documentation should prioritize clarity by using simple language, visual aids, and interactive tutorials. It should be comprehensive, detailing all features, including troubleshooting sections, and providing real-world examples. The documentation should be accessible in multiple formats and languages with intuitive navigation. It should also remain current with version control, maintain archives for system evolution, and actively promote community engagement through feedback platforms and valuing contributions.

C. OTHER TOPICS

1) ARCHITECTURAL STYLE

Different architectural styles are supported by smart city middleware. The Service-Oriented Architecture promotes seamless integration and communication between loosely-coupled, reusable services. Distributed architecture facilitates decentralized processing, allowing components to run concurrently across various locations for improved performance and reliability. Microservices architecture breaks down functionalities into small, independent services, each handling specific tasks, enabling flexibility and scaling.

Agent-based architecture utilizes autonomous agents that independently perform tasks, interact with each other, and adapt based on the environment, optimizing complex operations. Blockchain-based architecture offers a secure and transparent approach to data management by utilizing decentralized ledgers that ensure data integrity and trust. Lastly, hybrid architecture combines multiple architectural styles, leveraging the strengths of each to create a versatile and robust system.

2) PROGRAMMING PARADIGM

Smart city middleware utilizes diverse programming paradigms, each enhancing application development. Service-oriented paradigms ensure smooth communication through reusable services. Data-centric approaches prioritize efficient data handling. Event-driven models respond to specific urban triggers, and mashup paradigms enable the combination of different data sources and services to create comprehensive and user-friendly applications.

3) QUALITY OF SERVICE

For optimal service quality, middleware should employ scalable algorithms for real-time data, prioritize emergencies, optimize networks, and collaborate with urban experts. Dynamic resource management is essential and can be realized through predictive modeling, edge and fog computing, and fault tolerance. Furthermore, middleware must uphold Service Level Agreements, integrating AI and ML for breach predictions and collaborating with stakeholders to tailor requirements.

4) ENERGY EFFICIENCY

In achieving energy efficiency, smart city middleware can adopt three key strategies. First is adaptive data transmission, which utilizes algorithms and machine learning to set frequencies based on data significance, network conditions, and energy, all while collaborating with device makers for refinements. Second is dynamic energy management, emphasizing real-time device monitoring, energy-centric task scheduling, efficient communication protocols, and the leveraging of energy harvesting to prolong IoT device lifespan. Lastly, edge computing and offloading enhance energy efficiency through efficient task reallocation, by assessing the energy implications of processing decisions, and by collaboratively designing energy-efficient solutions with experts.

5) EVOLUTION AND MAINTENANCE

Middleware evolution highlights several key strategic areas. Emphasizing a modular design is crucial, achieved via microservices architecture, clear module interfaces, open standards, and efficient versioning. Equally vital are scalability and interoperability, ensured through adaptable components, universal protocols, automated resource allocation, and collaboration with stakeholders. For seamless updates, efficient patch systems, rollback capabilities, realistic testing environments, and clear user communication are essential. Additionally, the importance of continuous monitoring and feedback loops is highlighted by the use of advanced tools and periodic reviews, ensuring middleware remains attuned to evolving smart city needs and technological advancements.

VIII. DISCUSSION

A. INSIGHTS

This survey explores the fundamental characteristics of middleware platforms tailored for smart city applications. Insights derived from the results are presented, emphasizing their significance for various stakeholders in the smart city landscape, including middleware and application developers, administrators, operators, citizens, and academics.

In this research, various methodologies for the development of middleware platforms were systematically examined. The analysis conducted identified detailed functional requirements that can be further grouped into five broad categories. Firstly, *Data Management* stands out as a pivotal aspect, encompassing data collection, integration, storage, retrieval, processing, and analytics. This is critical given the vast amounts of heterogeneous data smart cities generate. Secondly, *Device and Infrastructure Management* addresses the need to manage IoT devices and allocate resources efficiently, ensuring robust connectivity and infrastructure reliability. Thirdly, *Service Management* is fundamental for provisioning, orchestrating, and maintaining various services within the smart city ecosystem. The fourth category, *Contextual and Event-Driven Processes*, emphasizes the importance of context-aware services and real-time responses, enabling middleware to react promptly to urban events and alerts. Lastly, the distinct category of *Application Lifecycle* is highlighted, underlining the comprehensive role middleware plays in supporting the entire life cycle of smart city applications, from initial development to deployment and ongoing maintenance.

For data management and processing in smart cities, Big Data, Cloud Computing and Semantic Web and Ontologies serve as pivotal technologies, facilitating large-scale data processing, scalable storage, and standardized data representation, while emphasizing scalability, interoperability and real-time data handling. For device and resource management, the IoT, Edge/Fog Computing, and CPS are pivotal; while IoT facilitates diverse device connectivity, Edge/Fog Computing focuses on localized data processing, and CPS integrates physical and software components, collectively emphasizing system configurability and adaptability to the dynamic device setting. For service management, Cloud Computing offers diverse hosting capabilities, AI drives enhanced ecosystem services, and Blockchain automates processes through smart contracts, ensuring efficient and reliable operations. In context and event management, Semantic Web and Ontologies, coupled with AI, enhance the understanding and interpretation of contextual data, enabling middleware to provide more precise, real-time responses based on rich, semantic insights and intelligent event processing. Cybersecurity is vital for all aspects, but it is especially crucial for

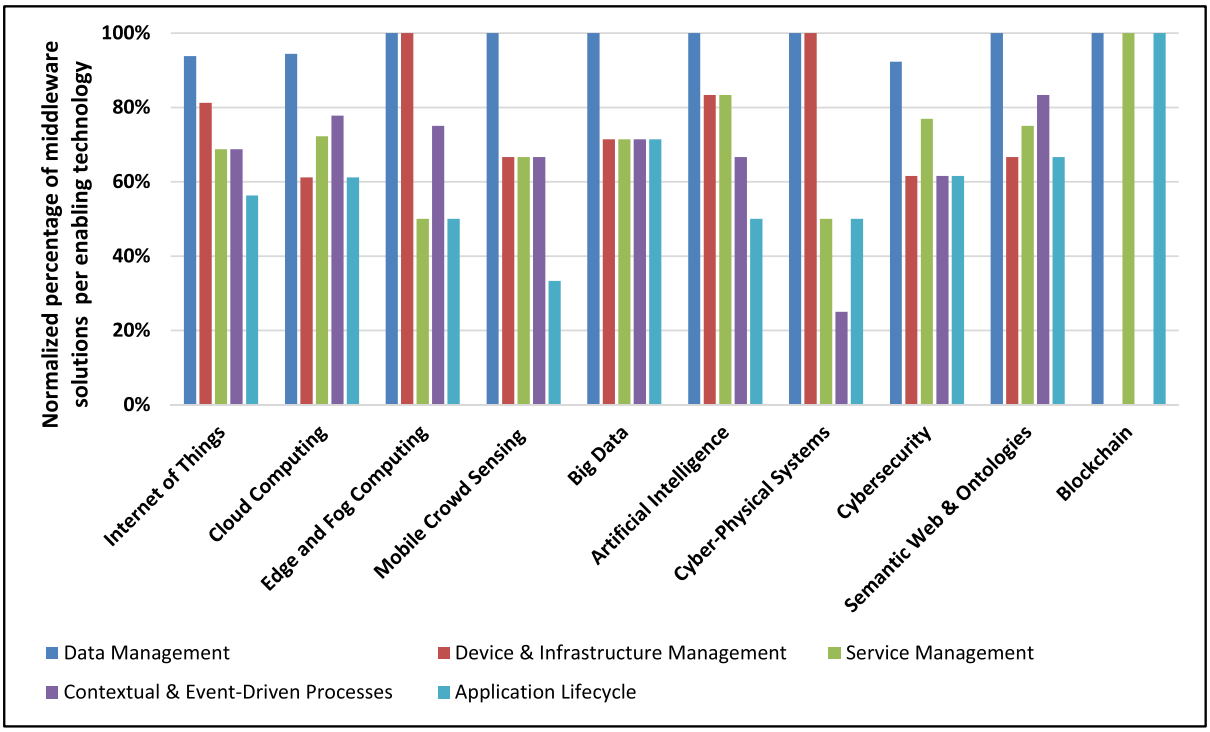

FIGURE 3. Relationship between enabling technologies and functional categories.

data and service management to ensure the protection of data and services.

Figure 3 depicts the relationship between enabling technologies and the functional categories they support, based on the sampled middleware solutions. The graph quantifies, in normalized percentages, the prevalence of each functional category within middleware solutions that adopt a specific enabling technology. For example, it shows that most of the solutions enabled by Semantic Web and Ontologies handle contextual and event-driven processes. On the other hand, device management is closely associated with Edge/Fog Computing and CPS. Through this graph, readers can discern which functional categories are predominantly associated with or influenced by particular enabling technologies in the current smart city middleware landscape. This visual representation offers insights into the prevailing trends, aiding stakeholders in understanding the intersection of enabling technologies and middleware functionalities.

The adoption of different architectural styles and programming paradigms in middleware solutions (Figure 4) provides insights into the evolving nature of smart city development. Predominance of Distributed and Service-Oriented architectures highlights the importance of decentralized, scalable systems that prioritize integration. The presence of Agent-Based and Microservices architectures reflects a trend towards modular, flexible, and adaptive systems. The emerging interest in Blockchain-Based architecture hints at an increasing emphasis on security, transparency, and trust in future smart city middleware. The preference towards Hybrid architectures, combining the benefits of two or more styles, indicates a strategic move to harness the unique strengths of diverse paradigms for optimal results.

On the programming front, the popularity of Service-Oriented approaches paired with Event-Driven and Data-Centric models reveals the emphasis on responsive, data-driven middleware platforms. This alignment is vital for handling the dynamic and diverse data sources of contemporary cities. The adoption of the Mashup model in specific platforms highlights the growing recognition that a cohesive synthesis of varied data streams and services can significantly enhance the richness and utility of smart city applications.

Figure 5 displays the mapping of smart city domains across the middleware solutions examined. While many popular sectors (such as smart mobility, home, energy, health, etc.) are represented, the percentages just indicate the frequency with which the explored middleware solutions were used to demonstrate their respective functionalities.

The plethora of enabling technologies identified, ranging from the IoT to Blockchain, heralds a transformative phase for the smart city ecosystem. For middleware and application developers, these technologies present an expansive toolbox, fostering innovation and enhancing adaptability in city systems. For example, given the data intensity of smart cities, developers should be well-acquainted with Big Data and Cloud Computing for efficient data storage and seamless network infrastructure. This understanding aids them in building resilient, dynamic, and efficient applications tailored to the unique demands of smart city ecosystems. For

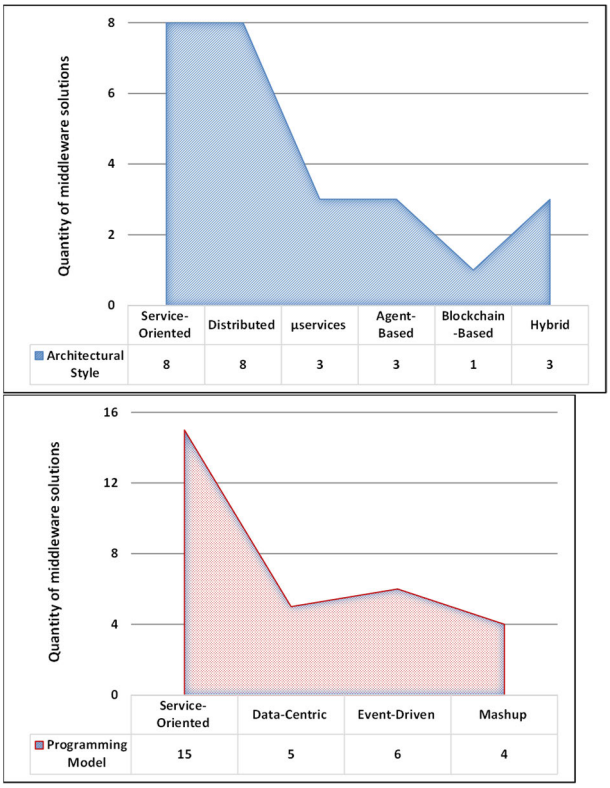

FIGURE 4. Representation of different architectural styles and programming models in the research sample.

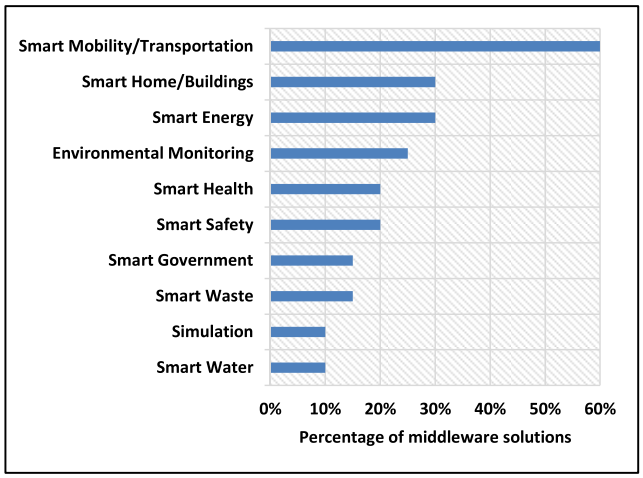

FIGURE 5. Mapping of smart city domains in the explored middleware solutions.

city administrators, these technologies provide a roadmap for the prospective evolution of the city, and understanding them can guide investment decisions. Operators, tasked with the day-to-day management and maintenance of urban systems, stand to gain substantially from these technological advancements. For instance, cybersecurity ensures that the infrastructure they oversee remains protected from external threats, ensuring the continuous and unhindered flow of urban operations. Citizens can expect more responsive, tailored, and secure services, with technologies like Semantic Web and Ontologies personalizing their urban experiences. Additionally, academics are presented with a rich field of interdisciplinary research opportunities, exploring the intersections, synergies, and challenges these technologies bring into the urban landscape.

The conceptual framework presented in this study features both functional and non-functional requirements integral to the development of middleware solutions tailored for smart cities. For middleware developers, our survey emphasizes the significance of a coordinated approach to data integration, management, and utilization. Application developers, guided by this conceptual framework, gain clarity on the array of services and data at their disposal for crafting applications suited to end-users. By highlighting application examples developed using the surveyed middleware solutions, the aim is to inform citizens about the diverse domains that aim to enhance their urban experience.

The open challenges presented in the survey can act as a roadmap for researchers, indicating the gaps in current knowledge and technology. This can guide them in formulating research questions, hypotheses, and experiments. Many challenges in smart city middleware, such as data privacy or IoT integration, may require interdisciplinary solutions. Understanding these challenges can encourage collaborations between computer scientists, urban planners, social scientists, and other specialists. Furthermore, the findings can be integrated into academic curricula to better prepare the next generation of middleware developers.

Middleware platforms will also shape co-creation as demonstrated by the WeValue platform [49]. Co-creation, in essence, is an initiative that stimulates various parties to collaboratively envision and forge a solution that holds mutual value. It transcends traditional development paradigms by actively involving end-users and other stakeholders right from the project's inception, thereby ensuring solutions that are more aligned with real-world needs and contexts. Therefore, co-creation forms a dynamic confluence of diverse actors including citizens and their communities, private sector entities, academic and knowledge institutes, and public organizations, each contributing their unique perspectives, insights, and expertise. A natural evolution of this is the emergence of e-Democracy, where digital tools and platforms, underpinned by technologies like Blockchain, empower citizens to actively partake in democratic processes [122]. Middleware platforms that integrate such technologies can potentially redefine the landscape of co-creation in smart cities, ensuring that every stakeholder, especially the citizens, have a transparent and influential voice in the evolution of their cities.

B. LIMITATIONS

In the analysis, middleware features and requirements were extracted from the most cited article of each examined solution and the respective project websites. It should be

acknowledged that other articles might present different features. Moreover, requirements, technologies, and other characteristics explored in the survey were identified as features in middleware solutions when they were explicitly or implicitly mentioned. If certain characteristics were cited as motivating topics or earmarked for future work, they were not included. For instance, InterSCity mentioned sophisticated Big Data processing and analytics along with improved data visualization as areas of future work [35]; therefore, these corresponding features were marked as 'not provided'.

The survey's insights, while valuable, are based on existing literature and middleware solutions. The rapidly evolving nature of technology means that even recent advancements might not be represented in the dataset if they have not been extensively cited or documented. Hence, while the survey provides a comprehensive overview of the middleware landscape, it might not offer a real-time snapshot of the very latest developments.

The success of middleware solutions also depends on factors external to their design, such as governance strategies, financial investments, citizen engagement, and social innovation initiatives [123]. The survey may not fully delve into these external factors. Economic considerations might deal with the distribution of resources, investment priorities, or the economic sustainability of certain solutions. By primarily spotlighting the technological solutions, there is a risk of developing a skewed perspective that does not holistically account for the multi-dimensional complexities inherent in the evolution of smart city middleware.

IX. CONCLUSION

As smart cities grow towards digital integration and intelligence, the role of middleware emerges as a critical facilitator, bridging the gap between diverse systems and ensuring seamless interoperability. In this survey, 20 middleware solutions were systematically examined to comprehend the benefits of employing a unified software platform over disjointed approaches and to pinpoint the crucial requirements for supporting the development, integration, and deployment of smart city applications. From the solutions analyzed, a conceptual framework for smart city middleware was formulated, offering a comprehensive guide for smart city stakeholders with considerations involving both functional and non-functional requirements. The formulated conceptual framework stems from the insights gathered through addressing the four research questions. Addressing RQ1, predominant enabling technologies for contemporary smart city middleware platforms were discerned. Subsequently, in response to RQ2, pertinent requirements intrinsic to smart city middleware were identified. Moreover, a mapping of the enabling technologies, aligning them with specific functionalities inherent to middleware platforms for smart cities was discussed. In addressing RQ3, a rigorous analysis of the foundational design principles governing smart city middleware development was conducted, alongside an examination of the programming paradigms that facilitate application development. Finally, in addressing RQ4, which probes into the major problems and emergent research issues concerning the development of smart city middleware, the challenges most frequently cited in academic literature were delineated. Consequently, a contribution of this study, particularly beneficial for middleware developers and researchers, lies in pinpointing the dimensions warranting focused exploration in future endeavors.

REFERENCES

[1] S. E. Bibri and J. Krogstie, "Smart sustainable cities of the future: An extensive interdisciplinary literature review," *Sustain. Cities Soc.*, vol. 31, pp. 183–212, May 2017.
[2] D. Gavalas, P. Nicopolitidis, A. Kameas, C. Goumopoulos, P. Bellavista, L. Lambrinos, and B. Guo, "Smart cities: Recent trends, methodologies, and applications," *Wireless Commun. Mobile Comput.*, vol. 2017, pp. 1–2, Oct. 2017.
[3] T. Boyle, D. Giurco, P. Mukheibir, A. Liu, C. Moy, S. White, and R. Stewart, "Intelligent metering for urban water: A review," *Water*, vol. 5, no. 3, pp. 1052–1081, Jul. 2013.
[4] A. Pitì, G. Verticale, C. Rottondi, A. Capone, and L. Lo Schiavo, "The role of smart meters in enabling real-time energy services for households: The Italian case," *Energies*, vol. 10, no. 2, p. 199, Feb. 2017.
[5] Z. Mahrez, E. Sabir, E. Badidi, W. Saad, and M. Sadik, "Smart urban mobility: When mobility systems meet smart data," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 6222–6239, Jul. 2022.
[6] E. Mbunge, B. Muchemwa, S. Jiyane, and J. Batani, "Sensors and healthcare 5.0: Transformative shift in virtual care through emerging digital health technologies," *Global Health J.*, vol. 5, no. 4, pp. 169–177, Dec. 2021.
[7] S. E. Bibri and J. Krogstie, "Environmentally data-driven smart sustainable cities: Applied innovative solutions for energy efficiency, pollution reduction, and urban metabolism," *Energy Informat.*, vol. 3, no. 1, pp. 1–59, Nov. 2020.
[8] S. Tulumello and F. Iapaolo, "Policing the future, disrupting urban policy today. Predictive policing, smart city, and urban policy in memphis (TN)," *Urban Geography*, vol. 43, no. 3, pp. 448–469, Feb. 2021.
[9] Z. Y. Dong, Y. Zhang, C. Yip, S. Swift, and K. Beswick, "Smart campus: Definition, framework, technologies, and services," *IET Smart Cities*, vol. 2, no. 1, pp. 43–54, Mar. 2020.
[10] A. Khanna, A. Sah, V. Bolshev, M. Jasinski, A. Vinogradov, Z. Leonowicz, and M. Jasiński, "Blockchain: Future of e-governance in smart cities," *Sustainability*, vol. 13, no. 21, Oct. 2021, Art. no. 11840.
[11] N. Tcholtchev and I. Schieferdecker, "Sustainable and reliable information and communication technology for resilient smart cities," *Smart Cities*, vol. 4, no. 1, pp. 156–176, Jan. 2021.
[12] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for smart cities," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 22–32, Feb. 2014.
[13] R. P. França, A. C. B. Monteiro, R. Arthur, and Y. Iano, "An overview of the machine learning applied in smart cities," in *Smart Cities: A Data Analytics Perspective*, M. A. Khan, F. Algarni, and M. T. Quasim, Eds. Cham, Switzerland: Springer, 2021, pp. 91–111.
[14] E. Badidi, Z. Mahrez, and E. Sabir, "Fog computing for smart cities' big data management and analytics: A review," *Future Internet*, vol. 12, no. 11, p. 190, Oct. 2020.
[15] T. Singh, A. Solanki, S. K. Sharma, A. Nayyar, and A. Paul, "A decade review on smart cities: Paradigms, challenges and opportunities," *IEEE Access*, vol. 10, pp. 68319–68364, 2022.
[16] M. Ibrahim, A. El-Zaart, and C. Adams, "Smart sustainable cities roadmap: Readiness for transformation towards urban sustainability," *Sustain. Cities Soc.*, vol. 37, pp. 530–540, Feb. 2018.
[17] E. F. Z. Santana, A. P. Chaves, M. A. Gerosa, F. Kon, and D. S. Milojicic, "Software platforms for smart cities: Concepts, requirements, challenges, and a unified reference architecture," *ACM Comput. Surv.*, vol. 50, no. 6, pp. 1–37, Nov. 2017.
[18] J. Pereira, T. Batista, E. Cavalcante, A. Souza, F. Lopes, and N. Cacho, "A platform for integrating heterogeneous data and developing smart city applications," *Future Gener. Comput. Syst.*, vol. 128, pp. 552–566, Mar. 2022.

[19] A. Pliatsios, K. Kotis, and C. Goumopoulos, "A systematic review on semantic interoperability in the IoE-enabled smart cities," *Internet Things*, vol. 22, Jul. 2023, Art. no. 100754.

[20] M. H. P. Rizi and S. A. H. Seno, "A systematic review of technologies and solutions to improve security and privacy protection of citizens in the smart city," *Internet Things*, vol. 20, Nov. 2022, Art. no. 100584.

[21] M. A. Razzaque, M. Milojevic-Jevric, A. Palade, and S. Clarke, "Middleware for Internet of Things: A survey," *IEEE Internet Things J.*, vol. 3, no. 1, pp. 70–95, Feb. 2016.

[22] S. Ketu and P. K. Mishra, "A contemporary survey on IoT based smart cities: Architecture, applications, and open issues," *Wireless Pers. Commun.*, vol. 125, no. 3, pp. 2319–2367, Mar. 2022.

[23] J. J. P. Abadía, C. Walther, A. Osman, and K. Smarsly, "A systematic survey of Internet of Things frameworks for smart city applications," *Sustain. Cities Soc.*, vol. 83, Aug. 2022, Art. no. 103949.

[24] K. Astropekakis, E. Drakakis, K. Grammatikakis, and C. Goumopoulos, "A survey of IoT software platforms," in *Advances in Computing, Informatics, Networking and Cybersecurity*, P. Nicopolitidis, S. Misra, L. T. Yang, B. Zeigler, and Z. Ning, Eds. Cham, Switzerland: Springer, 2022, pp. 299–326.

[25] A. Farahzadi, P. Shams, J. Rezazadeh, and R. Farahbakhsh, "Middleware technologies for cloud of things: A survey," *Digit. Commun. Netw.*, vol. 4, no. 3, pp. 176–188, Aug. 2018.

[26] N. Alasbali, S. R. Azzuhri, R. Salleh, and M. L. M. Kiah, "Blockchain as a middleware for IoT sensing and authentication within smart cities: A systematic literature review," *Res. Square*, vol. 1, pp. 1–42, Dec. 2020.

[27] K.-C. Chang, K.-C. Chu, H.-C. Wang, Y.-C. Lin, and J.-S. Pan, "Agent-based middleware framework using distributed CPS for improving resource utilization in smart city," *Future Gener. Comput. Syst.*, vol. 108, pp. 445–453, Jul. 2020.

[28] D. Puiu, P. Barnaghi, R. Tönjes, D. Kümper, M. I. Ali, A. Mileo, J. X. Parreira, M. Fischer, S. Kolozali, N. Farajidavar, F. Gao, T. Iggena, T.-L. Pham, C.-S. Nechifor, D. Puschmann, and J. Fernandes, "CityPulse: Large scale data analytics framework for smart cities," *IEEE Access*, vol. 4, pp. 1086–1108, 2016.

[29] F. J. Villanueva, M. J. Santofimia, D. Villa, J. Barba, and J. C. López, "Civitas: The smart city middleware, from sensors to big data," in *Proc. 7th Int. Conf. Innov. Mobile Internet Services Ubiquitous Comput.*, Jul. 2013, pp. 445–450.

[30] F. Cirillo, G. Solmaz, E. L. Berz, M. Bauer, B. Cheng, and E. Kovacs, "A standard-based open source IoT platform: FIWARE," *IEEE Internet Things Mag.*, vol. 2, no. 3, pp. 12–18, Sep. 2019.

[31] O. M. Alliance. (May 2010). *Next Generation Service Interfaces Architecture*. [Online]. Available: https://www.openmobilealliance.org/release/NGSI/V1_0-20101207-C/OMA-AD-NGSI-V1_0-20100803-C.pdf

[32] ETSI. (2019). *Context Information Management (CIM); NGSI-LD API*. [Online]. Available: http://bit.ly/2LJxul6

[33] B. Cheng, G. Solmaz, F. Cirillo, E. Kovacs, K. Terasawa, and A. Kitazawa, "FogFlow: Easy programming of IoT services over cloud and edges for smart cities," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 696–707, Apr. 2018.

[34] W. Apolinarski, U. Iqbal, and J. X. Parreira, "The GAMBAS middleware and SDK for smart city applications," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PERCOM WORKSHOPS)*, Mar. 2014, pp. 117–122.

[35] A. M. Del Esposte, F. Kon, F. M. Costa, and N. Lago, "InterSCity: A scalable microservice-based open source platform for smart cities," in *Proc. 6th Int. Conf. Smart Cities Green ICT Syst.*, Apr. 2017, pp. 35–46.

[36] E. Patti, A. Acquaviva, M. Jahn, F. Pramudianto, R. Tomasi, D. Rabourdin, J. Virgone, and E. Macii, "Event-driven user-centric middleware for energy-efficient buildings and public spaces," *IEEE Syst. J.*, vol. 10, no. 3, pp. 1137–1146, Sep. 2016.

[37] J. Aguilar, M. Jerez, M. Mendonça, and M. Sánchez, "Performance analysis of the ubiquitous and emergent properties of an autonomic reflective middleware for smart cities," *Computing*, vol. 102, no. 10, pp. 2199–2228, Mar. 2020.

[38] J. Soldatos, "OpenIoT: Open source Internet-of-Things in the cloud," in *Interoperability and Open-Source Solutions for the Internet of Things*, I. P. Žarko, K. Pripužić, and M. Serrano, Eds. Cham, Switzerland: Springer, 2015, pp. 13–25.

[39] M. Compton, "The SSN ontology of the W3C semantic sensor network incubator group," *J. Web Semantics*, vol. 17, pp. 25–32, Dec. 2012.

[40] C. Huo, T.-C. Chien, and P. H. Chou, "Middleware for IoT-cloud integration across application domains," *IEEE Des. Test.*, vol. 31, no. 3, pp. 21–31, Jun. 2014.

[41] A. Pliatsios, D. Lymperis, and C. Goumopoulos, "S2NetM: A semantic social network of things middleware for developing smart and collaborative IoT-based solutions," *Future Internet*, vol. 15, no. 6, p. 207, Jun. 2023.

[42] D. Lymperis and C. Goumopoulos, "SEDIA: A platform for semantically enriched IoT data integration and development of smart city applications," *Future Internet*, vol. 15, no. 8, p. 276, Aug. 2023.

[43] L. Dantas, E. Cavalcante, and T. Batista, "A development environment for FIWARE-based Internet of Things applications," in *Proc. 6th Int. Workshop Middleware Appl. Internet Things*, Dec. 2019, pp. 21–26.

[44] J. Rodríguez-Molina, J.-F. Martínez, P. Castillejo, and R. de Diego, "SMArc: A proposal for a smart, semantic middleware architecture focused on smart city energy management," *Int. J. Distrib. Sensor Netw.*, vol. 9, no. 12, Dec. 2013, Art. no. 560418.

[45] N. Mohamed, J. Al-Jaroodi, I. Jawhar, S. Lazarova-Molnar, and S. Mahmoud, "SmartCityWare: A service-oriented middleware for cloud and fog enabled smart city services," *IEEE Access*, vol. 5, pp. 17576–17588, 2017.

[46] L. Sanchez, L. Muñoz, J. A. Galache, P. Sotres, J. R. Santana, V. Gutierrez, R. Ramdhany, A. Gluhak, S. Krco, E. Theodoridis, and D. Pfisterer, "SmartSantander: IoT experimentation over a smart city testbed," *Comput. Netw.*, vol. 61, pp. 217–238, Mar. 2014.

[47] C. Badii, E. G. Belay, P. Bellini, D. Cenni, M. Marazzini, M. Mesiti, P. Nesi, G. Pantaleo, M. Paolucci, S. Valtolina, M. Soderi, and I. Zaza, "Snap4City: A scalable IoT/IOE platform for developing smart city applications," in *Proc. IEEE SmartWorld, Ubiquitous Intell. Comput., Adv. Trusted Comput., Scalable Comput. Commun., Cloud Big Data Comput., Internet People Smart City Innov. (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, Oct. 2018, pp. 2109–2116.

[48] H. S. Jung, C. S. Yoon, Y. Lee, J. W. Park, and C. H. Yun, "Processing IoT data with cloud computing for smart cities," *Int. J. Web Appl.*, vol. 9, no. 3, pp. 88–95, Sep. 2017.

[49] O. Scekic, S. Nastic, and S. Dustdar, "Blockchain-supported smart city platform for social value co-creation and exchange," *IEEE Internet Comput.*, vol. 23, no. 1, pp. 19–28, Jan. 2019.

[50] O. Scekic, T. Schiavinotto, S. Videnov, M. Rovatsos, H.-L. Truong, D. Miorandi, and S. Dustdar, "A programming model for hybrid collaborative adaptive systems," *IEEE Trans. Emerg. Topics Comput.*, vol. 8, no. 1, pp. 6–19, Jan. 2020.

[51] S. Bansal and D. Kumar, "IoT ecosystem: A survey on devices, gateways, operating systems, middleware and communication," *Int. J. Wireless Inf. Netw.*, vol. 27, no. 3, pp. 340–364, Feb. 2020.

[52] M. H. Amini, J. Mohammadi, and S. Kar, "Distributed holistic framework for smart city infrastructures: Tale of interdependent electrified transportation network and power grid," *IEEE Access*, vol. 7, pp. 157535–157554, 2019.

[53] T. P. Da Silva, T. Batista, F. Lopes, A. R. Neto, F. C. Delicato, P. F. Pires, and A. R. Da Rocha, "Fog computing platforms for smart city applications: A survey," *ACM Trans. Internet Technol.*, vol. 22, no. 4, pp. 1–32, Dec. 2022.

[54] T. Alam, "Cloud-based IoT applications and their roles in smart cities," *Smart Cities*, vol. 4, no. 3, pp. 1196–1219, Sep. 2021.

[55] K. Tei, L. Güreen, and T. Yonezawa, "ClouT: Cloud of things for empowering citizen's clout in smart cities," in *Enablers for Smart Cities*, A. E. F. Seghrouchni, F. Ishikawa, L. Herault, and H. Tokuda, Eds., Hoboken, NJ, USA: Wiley, 2016, pp. 107–126.

[56] A. Sunyaev, "Fog and edge computing," in *Internet Computing: Principles of Distributed Systems and Emerging Internet-Based Technologies*, A. Sunyaev, Ed. Cham, Switzerland: Springer, 2020, pp. 237–264.

[57] L. U. Khan, I. Yaqoob, N. H. Tran, S. M. A. Kazmi, T. N. Dang, and C. S. Hong, "Edge-computing-enabled smart cities: A comprehensive survey," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 10200–10232, Oct. 2020.

[58] B. Guo, Z. Wang, Z. Yu, Y. Wang, N. Y. Yen, R. Huang, and X. Zhou, "Mobile crowd sensing and computing: The review of an emerging human-powered sensing paradigm," *ACM Comput. Surv.*, vol. 48, no. 1, pp. 1–31, Aug. 2015.

[59] H. Habibzadeh, C. Kaptan, T. Soyata, B. Kantarci, and A. Boukerche, "Smart city system design: A comprehensive study of the application and data planes," *ACM Comput. Surv.*, vol. 52, no. 2, pp. 1–38, May 2019.
[60] A. Capponi, C. Fiandrino, B. Kantarci, L. Foschini, D. Kliazovich, and P. Bouvry, "A survey on mobile crowdsensing systems: Challenges, solutions, and opportunities," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2419–2465, 3rd Quart., 2019.
[61] M. Hirsch, C. Mateos, and A. Zunino, "Augmenting computing capabilities at the edge by jointly exploiting mobile devices: A survey," *Future Gener. Comput. Syst.*, vol. 88, pp. 644–662, Nov. 2018.
[62] H. R. Arkian, A. Diyanat, and A. Pourkhalili, "MIST: Fog-based data analytics scheme with cost-efficient resource provisioning for IoT crowdsensing applications," *J. Netw. Comput. Appl.*, vol. 82, pp. 152–165, Mar. 2017.
[63] I. A. T. Hashem, "The role of big data in smart city," *Int. J. Inf. Manage.*, vol. 36, no. 5, pp. 748–758, 2016.
[64] H. Özköse, E. S. Arı, and C. Gencer, "Yesterday, today and tomorrow of big data," *Proc. Social Behav. Sci.*, vol. 195, pp. 1042–1050, Jul. 2015.
[65] Z. Chaczko, R. Klempous, J. Rozenblit, C. Chiu, K. Kluwak, and C. Smutnicki, "Enabling design of middleware for massive scale IoT-based systems," in *Proc. IEEE 23rd Int. Conf. Intell. Eng. Syst. (INES)*, Apr. 2019, pp. 219–223.
[66] J. Wang, Y. Yang, T. Wang, R. S. Sherratt, and J. Zhang, "Big data service architecture: A survey," *J. Internet Technol.*, vol. 21, no. 2, pp. 393–405, 2020.
[67] L. Batagan, "Open data for smart cities," *ASE Economy Inform.*, vol. 12, no. 1, p. 136, 2012.
[68] F. T. Neves, M. de Castro Neto, and M. Aparicio, "The impacts of open data initiatives on smart cities: A framework for evaluation and monitoring," *Cities*, vol. 106, Nov. 2020, Art. no. 102860.
[69] A. R. Javed, W. Ahmed, S. Pandya, P. K. R. Maddikunta, M. Alazab, and T. R. Gadekallu, "A survey of explainable artificial intelligence for smart cities," *Electronics*, vol. 12, no. 4, p. 1020, Feb. 2023.
[70] T. Bakhshi and M. Ahmed, "IoT-enabled smart city waste management using machine learning analytics," in *Proc. 2nd Int. Conf. Energy Conservation Efficiency (ICECE)*, Oct. 2018, pp. 66–71.
[71] M. S. Hossain, G. Muhammad, and N. Guizani, "Explainable AI and mass surveillance system-based healthcare framework to combat COVID-I9 like pandemics," *IEEE Netw.*, vol. 34, no. 4, pp. 126–132, Jul. 2020.
[72] R. Baheti and H. Gill, "Cyber-physical systems," in *The Impact of Control Technology*, T. Samad, and A. M. Annaswamy, Eds., New York, NY, USA: IEEE Control Systems Society, 2011, pp. 161–166.
[73] M. Juma and K. Shaalan, "Cyberphysical systems in the smart city: Challenges and future trends for strategic research," in *Swarm Intelligence for Resource Management in Internet of Things*, A. E. Hassanien, and A. Darwish, Eds. London, U.K.: Academic, 2020, pp. 65–85.
[74] A. Rasheed, O. San, and T. Kvamsdal, "Digital twin: Values, challenges and enablers from a modeling perspective," *IEEE Access*, vol. 8, pp. 21980–22012, 2020.
[75] G. Mylonas, A. Kalogeras, G. Kalogeras, C. Anagnostopoulos, C. Alexakos, and L. Muñoz, "Digital twins from smart manufacturing to smart cities: A survey," *IEEE Access*, vol. 9, pp. 143222–143249, 2021.
[76] F. Tao, Q. Qi, L. Wang, and A. Y. C. Nee, "Digital twins and cyber–physical systems toward smart manufacturing and Industry 4.0: Correlation and comparison," *Engineering*, vol. 5, no. 4, pp. 653–661, Aug. 2019.
[77] H. Habibzadeh, B. H. Nussbaum, F. Anjomshoa, B. Kantarci, and T. Soyata, "A survey on cybersecurity, data privacy, and policy issues in cyber-physical system deployments in smart cities," *Sustain. Cities Soc.*, vol. 50, Oct. 2019, Art. no. 101660.
[78] V. Wylde, N. Rawindaran, J. Lawrence, R. Balasubramanian, E. Prakash, A. Jayal, I. Khan, C. Hewage, and J. Platts, "Cybersecurity, data privacy and blockchain: A review," *Social Netw. Comput. Sci.*, vol. 3, no. 2, p. 127, Jan. 2022.
[79] M. Z. Gunduz and R. Das, "Cyber-security on smart grid: Threats and potential solutions," *Comput. Netw.*, vol. 169, Mar. 2020, Art. no. 107094.
[80] H. F. Atlam and G. B. Wills, "IoT security, privacy, safety and ethics," in *Digital Twin Technologies and Smart Cities*, M. Farsi, A. Daneshkhah, A. Hosseinian-Far, and H. Jahankhani, Eds. Cham, Switzerland: Springer, 2020, pp. 123–149.
[81] G. Ahmadi-Assalemi, H. Al-Khateeb, G. Epiphaniou, and C. Maple, "Cyber resilience and incident response in smart cities: A systematic literature review," *Smart Cities*, vol. 3, no. 3, pp. 894–927, Aug. 2020.
[82] S. Atkins and C. Lawson, "An improvised patchwork: Success and failure in cybersecurity policy for critical infrastructure," *Public Admin. Rev.*, vol. 81, no. 5, pp. 847–861, Sep. 2021.
[83] A. Pliatsios, C. Goumopoulos, and K. Kotis, "A review on IoT frameworks supporting multi-level interoperability-the semantic social network of things framework," *Int. J. Adv. Internet Technol.*, vol. 13, no. 1, pp. 46–64, 2020.
[84] B. Bhushan, A. Khamparia, K. M. Sagayam, S. K. Sharma, M. A. Ahad, and N. C. Debnath, "Blockchain for smart cities: A review of architectures, integration trends and future research directions," *Sustain. Cities Soc.*, vol. 61, Oct. 2020, Art. no. 102360.
[85] B. Bhushan, C. Sahoo, P. Sinha, and A. Khamparia, "Unification of blockchain and Internet of Things (BIoT): Requirements, working model, challenges and future directions," *Wireless Netw.*, vol. 27, no. 1, pp. 55–90, Jan. 2021.
[86] C. Gao, Y. Ji, J. Wang, and X. Sai, "Application of blockchain technology in peer-to-peer transaction of photovoltaic power generation," in *Proc. 2nd IEEE Adv. Inf. Manag.,Communicates,Electron. Autom. Control Conf. (IMCEC)*, May 2018, pp. 2289–2293.
[87] Z. Li, S. Bahramirad, A. Paaso, M. Yan, and M. Shahidehpour, "Blockchain for decentralized transactive energy management system in networked microgrids," *Electr. J.*, vol. 32, no. 4, pp. 58–72, May 2019.
[88] G. Malik, K. Parasrampuria, S. P. Reddy, and S. Shah, "Blockchain based identity verification model," in *Proc. Int. Conf. Vis. Towards Emerg. Trends Commun. Netw. (ViTECoN)*, Mar. 2019, pp. 1–6.
[89] M. Montecchi, K. Plangger, and M. Etter, "It's real, trust me! Establishing supply chain provenance using blockchain," *Bus. Horizons*, vol. 62, no. 3, pp. 283–293, May 2019.
[90] J. Thönes, "Microservices," *IEEE Softw.*, vol. 32, no. 1, p. 116, Jan. 2015.
[91] S. Abar, G. K. Theodoropoulos, P. Lemarinier, and G. M. P. O'Hare, "Agent based modelling and simulation tools: A review of the state-of-art software," *Comput. Sci. Rev.*, vol. 24, pp. 13–33, May 2017.
[92] S. N. Swamy and S. R. Kota, "An empirical study on system level aspects of Internet of Things (IoT)," *IEEE Access*, vol. 8, pp. 188082–188134, 2020.
[93] P. A. Alvaro, "Data-centric programming for distributed systems," Ph.D. dissertation, Dept. Elect. Eng. Comput. Sci., Univ. California, Berkeley, CA, USA, 2015. [Online]. Available: https://escholarship.org/
[94] F. Dabek, N. Zeldovich, F. Kaashoek, D. Mazières, and R. Morris, "Event-driven programming for robust software," in *Proc. 10th workshop ACM SIGOPS Eur. Workshop, Beyond PC (EW)*, Jul. 2002, pp. 186–189.
[95] J. Yu, B. Benatallah, F. Casati, and F. Daniel, "Understanding mashup development," *IEEE Internet Comput.*, vol. 12, no. 5, pp. 44–52, Sep. 2008.
[96] B. Ahlgren, M. Hidell, and E. C.-H. Ngai, "Internet of Things for smart cities: Interoperability and open data," *IEEE Internet Comput.*, vol. 20, no. 6, pp. 52–56, Nov. 2016.
[97] A. de M. Del Esposte, E. F. Z. Santana, L. Kanashiro, F. M. Costa, K. R. Braghetto, N. Lago, and F. Kon, "Design and evaluation of a scalable smart city software platform with large-scale simulations," *Future Gener. Comput. Syst.*, vol. 93, pp. 427–441, Apr. 2019.
[98] A. H. Sodhro, S. Pirbhulal, Z. Luo, and V. H. C. de Albuquerque, "Towards an optimal resource management for IoT based green and sustainable smart cities," *J. Cleaner Prod.*, vol. 220, pp. 1167–1179, May 2019.
[99] J. Mondschein, A. Clark-Ginsberg, and A. Kuehn, "Smart cities as large technological systems: Overcoming organizational challenges in smart cities through collective action," *Sustain. Cities Soc.*, vol. 67, Apr. 2021, Art. no. 102730.
[100] S. Jain, S. Gupta, K. K. Sreelakshmi, and J. J. P. C. Rodrigues, "Fog computing in enabling 5G-driven emerging technologies for development of sustainable smart city infrastructures," *Cluster Comput.*, vol. 25, no. 2, pp. 1111–1154, Jan. 2022.
[101] M. Hirsch, C. Mateos, A. Zunino, T. A. Majchrzak, T.-M. Grønli, and H. Kaindl, "A task execution scheme for dew computing with State-of-the-Art smartphones," *Electronics*, vol. 10, no. 16, p. 2006, Aug. 2021.
[102] A. Shahraki, M. Geitle, and Ø. Haugen, "A comparative node evaluation model for highly heterogeneous massive-scale Internet of Things-Mist networks," *Trans. Emerg. Telecommun. Technol.*, vol. 31, no. 12, Dec. 2020, Art. no. e3924.

[103] D. Eckhoff and I. Wagner, "Privacy in the smart city—Applications, technologies, challenges, and solutions," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 1, pp. 489–516, 1st Quart., 2018.
[104] N. Momen, "Privacy and ethics in a smart city: Towards attaining digital sovereignty," in *Cybersecurity for Smart Cities: Practices and Challenges*, M. Ahmed, and P. Haskell-Dowland, Eds. Cham, Switzerland: Springer, 2023, pp. 47–60.
[105] E. Ismagilova, L. Hughes, N. P. Rana, and Y. K. Dwivedi, "Security, privacy and risks within smart cities: Literature review and development of a smart city interaction framework," *Inf. Syst. Frontiers*, vol. 24, no. 2, pp. 393–414, Apr. 2022.
[106] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context aware computing for the Internet of Things: A survey," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 414–454, 1st Quart., 2014.
[107] P. Bellavista, A. Corradi, M. Fanelli, and L. Foschini, "A survey of context data distribution for mobile ubiquitous systems," *ACM Comput. Surv.*, vol. 44, no. 4, pp. 1–45, Sep. 2012.
[108] C. Lim, K.-J. Kim, and P. P. Maglio, "Smart cities with big data: Reference models, challenges, and considerations," *Cities*, vol. 82, pp. 86–99, Dec. 2018.
[109] S. E. Bibri, "A foundational framework for smart sustainable city development: Theoretical, disciplinary, and discursive dimensions and their synergies," *Sustain. Cities Soc.*, vol. 38, pp. 758–794, Apr. 2018.
[110] Y. A. Aina, "Achieving smart sustainable cities with GeoICT support: The Saudi evolving smart cities," *Cities*, vol. 71, pp. 49–58, Nov. 2017.
[111] U. Aguilera, O. Peña, O. Belmonte, and D. López-de-Ipiña, "Citizen-centric data services for smarter cities," *Future Gener. Comput. Syst.*, vol. 76, pp. 234–247, Nov. 2017.
[112] M. A. Mukwevho and T. Celik, "Toward a smart cloud: A review of fault-tolerance methods in cloud systems," *IEEE Trans. Services Comput.*, vol. 14, no. 2, pp. 589–605, Mar. 2021.
[113] R. Angarita, M. Rukoz, M. Manouvrier, and Y. Cardinale, "A knowledge-based approach for self-healing service-oriented applications," in *Proc. 8th Int. Conf. Manage. Digit. EcoSyst.*, Nov. 2016, pp. 1–8.
[114] C. Engelmann, G. R. Vallee, T. Naughton, and S. L. Scott, "Proactive fault tolerance using preemptive migration," in *Proc. 17th Euromicro Int. Conf. Parallel, Distrib. Netw.-Based Process.*, Feb. 2009, pp. 252–257.
[115] F. Salfner, M. Lenk, and M. Malek, "A survey of online failure prediction methods," *ACM Comput. Surv.*, vol. 42, no. 3, pp. 1–42, Mar. 2010.
[116] K. Nivitha and P. Pabitha, "A survey on machine learning based fault tolerant mechanisms in cloud towards uncertainty analysis," in *Proc. Int. Conf. Comput. Netw., Big Data IoT*, vol. 2020, pp. 13–20.
[117] W. Tu, "Data-driven QoS and QoE management in smart cities: A tutorial study," *IEEE Commun. Mag.*, vol. 56, no. 12, pp. 126–133, Dec. 2018.
[118] D. J. Ecklund, V. Goebel, T. Plagemann, and E. F. Ecklund Jr., "Dynamic end-to-end QoS management middleware for distributed multimedia systems," *Multimedia Syst.*, vol. 8, no. 5, pp. 431–442, Dec. 2002.
[119] O. A. Khashan, R. Ahmad, and N. M. Khafajah, "An automated lightweight encryption scheme for secure and energy-efficient communication in wireless sensor networks," *Ad Hoc Netw.*, vol. 115, Apr. 2021, Art. no. 102448.
[120] C. Jiang, T. Fan, H. Gao, W. Shi, L. Liu, C. Cérin, and J. Wan, "Energy aware edge computing: A survey," *Comput. Commun.*, vol. 151, pp. 556–580, Feb. 2020.
[121] J. Al-Jaroodi and N. Mohamed, "Service-oriented middleware: A survey," *J. Netw. Comput. Appl.*, vol. 35, no. 1, pp. 211–220, Jan. 2012.
[122] E. Pournaras, "Proof of witness presence: Blockchain consensus for augmented democracy in smart cities," *J. Parallel Distrib. Comput.*, vol. 145, pp. 160–175, Nov. 2020.
[123] M. Angelidou and A. Psaltoglou, "An empirical investigation of social innovation initiatives for sustainable urban development," *Sustain. Cities Soc.*, vol. 33, pp. 113–125, Aug. 2017.

CHRISTOS GOUMOPOULOS (Member, IEEE) received the Ph.D. degree in computer science from the University of Patras, in 2000. He is currently an Associate Professor with the Department of Information and Communications Systems Engineering, University of the Aegean. He is also the Director of the Postgraduate Program titled Internet of Things: Intelligent Environments in Next-Generation Networks and heads the Human-Centric Computing Group. He is also a Cooperating Professor with the Mobile and Pervasive Computing Systems Postgraduate Program, Hellenic Open University. He is the author of four course books in the fields of pervasive computing and the Internet of Things, which are used in the corresponding postgraduate programs. He has (co) authored over 100 scientific publications. His research interests include pervasive computing system architectures, design and programming of pervasive computing systems, programming models and frameworks, middleware design, the IoT platforms, semantic interoperability in the IoT, ontological knowledge representation, ambient assisted living systems, and precision agriculture. He is a TPC member of international conferences.