OpenCity: An Open Architecture Testbed for Smart Cities

Nasibeh Zohrabi[1], Patrick J. Martin[2], Murat Kuzlu[3], Lauren Linkous[2], Roja Eini[2], Adam Morrissett[2], Mostafa Zaman[2], Ashraf Tantawy[2], Oezguer Gueler[5], Maher Al Islam[2], Nathan Puryear[2], Halil Kalkavan[4], Jonathan Lundquist[2], Erwin Karincic[2], Sherif Abdelwahed[2]
[1]*Department of Engineering, Pennsylvania State University*, Media, PA 19063, USA
[2]*Electrical and Computer Engineering, Virginia Commonwealth University*, Richmond, VA 23284, USA
[3]*Electrical Engineering Technology, Old Dominion University*, Norfolk, VA 23529, USA
[4]*Department of Computer Science, Old Dominion University*, Norfolk, VA 23529, USA
[5]*eKare, Inc.,* Fairfax, VA 22031, USA
nmz5171@psu.edu, martinp@vcu.edu, mkuzlu@odu.edu, linkouslc@vcu.edu, einir@vcu.edu, morrissettal2@vcu.edu, zamanm@vcu.edu, amatantawy@vcu.edu, oguler@ekareinc.com, alislamm@vcu.edu, puryearna@vcu.edu, hkalk001@odu.edu, lundquistjd@vcu.edu, erwink@vcu.edu, sabdelwahed@vcu.edu

***Abstract*—This paper presents an open architecture testbed for smart cities, called OpenCity, which is hosted at Virginia Commonwealth University (VCU). The OpenCity platform consists of data collection and processing units, database management, distributed performance management algorithms, and real-time data visualization. This smart city testbed aims to support various educational and research activities related to smart city development. The testbed provides a near-real-life platform to allow students to learn about the unique features of smart cities and explore supporting technologies. In addition, it allows researchers to develop, deploy, and validate new techniques, tools, and technologies to support future smart city developments. The OpenCity platform will support various ongoing important research directions in smart cities, including smart homes and buildings, urban mobility, smart grid, and water management. In addition, it will be extendable to include other potential applications and components as needed. The testbed will be validated by developing and deploying a management system that focuses on users' experience and resource efficiency. The management system incorporates learning techniques and model-based predictive control approaches to take into account the current and future information of uncertain parameters as well as the subjective data (e.g., user-related data) in the design. The OpenCity management structure enables real-time control and monitoring of complex components in the testbed.**

***Index Terms*—Smart City, OpenCity, Data Analytics, Smart Buildings, Intelligent Transportation Systems**

I. Introduction

More than 50 percent of the world's population currently reside in urban areas [1], and by 2050, two-thirds of the human population are predicted to live in urban areas [2]. It is envisioned that the Internet of Things (IoT) will provide a real-time information stream for city managers and public service providers to better understand the strain placed on the city's infrastructure and resources. The IoT is a network of physical devices (e.g., home appliances, medical sensors, vehicles) embedded with sensing, actuation, computing, and communications capabilities that enable distributed monitoring and control to reach common goals. By building smart, IoT-based services that serve human users, smart cities will enhance the safety, wellness, and quality of life for their citizens [3].

In 2015, the US Government invested over $160 million in the "smart cities" initiative to improve the quality of life, economic competitiveness, and sustainability of future cities [4]. This initiative involves a number of fundamental research areas, such as smart buildings [5], [6], smart grid [7], intelligent transportation systems [8], smart water/waste management [9], smart healthcare [10], and cybersecurity [11]. One important research challenge is the development of assured, data-driven management systems that control and monitor the city's operations in a safe, secure, and reliable manner.

Smart cities are complex systems-of-systems that contain a multitude of hardware and software platforms. To better understand realistic, emergent behavior, a testbed that emulates several core smart city systems is needed. This paper introduces a smart city testbed, called *OpenCity*, which is hosted at Virginia Commonwealth University (VCU). *OpenCity* will provide a more cost-effective way for researchers to develop technologies and techniques to deliver optimized programs and services to smart city residents. The testbed will establish a standards-based framework and a physical, real-time experimental environment to develop and evaluate new decision-making and system management solutions.

In recent years, several other smart city and IoT testbeds were developed. For example, Smart Santander [12] implemented a full scale testbed that emphasizes mobile device management and data collection. The FIT IoT Lab [13] is deployed across multiple buildings and institutions. It provides users the ability to develop and deploy mobility applications at internet and city scale. Another recent testbed, City of Things [14], integrates multiple radio and protocols to enable researchers to experiment with city scale data and network management. The key difference between *OpenCity* and these platforms is the facilitation of integrated, command and control of smart city resources. For example, due to regulatory limitations, it is not possible to perform full scale tests of autonomous mobility applications; therefore, we require lab-scale testbeds to develop and evaluate decision support systems.

The main contribution of the *OpenCity* platform is to provide an accessible experimental environment for researchers to explore algorithms and technologies that will make intelligent, sustainable cities. This new testbed relies on IoT-based communication technologies, autonomous systems, software development, and database management to address the issue of plug-and-play, scalability, and interoperability. The project team will also develop a set of distributed management control algorithms that supports decision-making processes for efficient smart city operations. The proposed platform will allow researchers to collaborate and share data as well as grow a community of smart city enthusiasts that contributes to innovative smart city applications.

In the *OpenCity* platform, we focus on different applications to smart cities, including smart homes and buildings, and smart transportation systems. A data-driven management structure will be developed that supports autonomous and semi-autonomous operation of smart city infrastructure. Additionally, the proposed management structure will allow researchers to learn how to incorporate various human aspects into the management structure. This testbed supports educational and research activities by engaging with government and industry partners to accelerate understanding of analytics and smart technologies and transition research into real-life solutions.

The rest of the paper is organized as follows. Sections II and III introduce the *OpenCity* design layout and communication architecture. Section IV presents the design architecture of *OpenCity* intelligent transportation system. Section V presents the design considerations for four different types of *OpenCity* smart buildings. Section VI presents the *OpenCity* management architecture. Finally, the conclusion is given in Section VII.

II. OpenCity High Level Design

The *OpenCity* platform includes the subsystems illustrated in Fig. 1: smart residential and commercial buildings, intelligent transportation system, as well as the associated communication backbone and data analytics infrastructure. All entities in this smart city model are assumed to have an intelligent node that performs local sensing and control, or optionally uses control signals issued from edge or cloud management systems.

The goal of this testbed is to provide a smart city experimental environment that includes a) data collection, processing and analytics, b) an IoT communication network, c) real-time data visualization, and d) a configurable and extendable management system. Fig. 2 illustrates an *OpenCity* "block" that has four buildings placed in a two by two grid and surrounded on all sides by roads. A four-way intersection featuring multiple traffic lights runs through the center. These modular blocks may be connected with others to construct a larger city grid. The *OpenCity* block includes four distinct buildings: a) Residential, b) Commercial office, c) Hospital, and d) Water treatment plant. The transportation system will operate autonomous vehicles, road sensors, and traffic signals that may communicate with *OpenCity* infrastructure. The following sections describe the features and design considerations of each *OpenCity* subsystem.

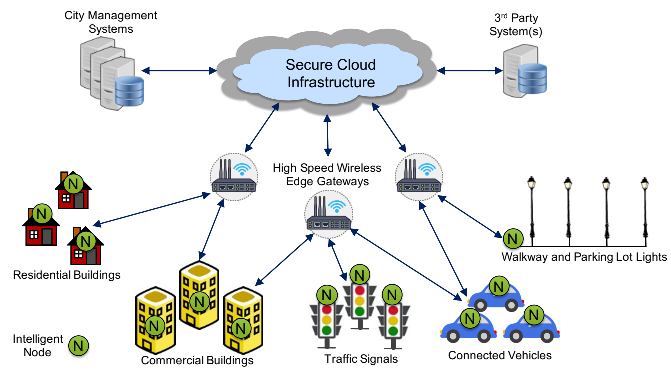


Fig. 1: An illustration of a typical Smart City computing architecture.

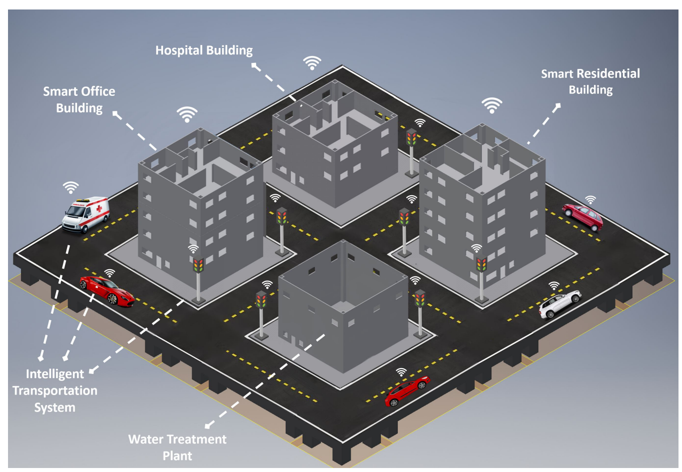


Fig. 2: The *OpenCity* platform design featuring four buildings, intersecting roads, and peripheral features for testing the intelligent transportation system.

III. OpenCity Communication Architecture

The *OpenCity* architecture, illustrated in Fig. 3, consists of decentralized software services and cyber-physical entities that are bound together with a common message transport protocol, Message Queuing Telemetry Transport (MQTT)[1]. The MQTT protocol is an OASIS standard and is heavily used in the commercial sector for smart city applications. The loose coupling of all systems with a common publication/subscribe messaging system allows *OpenCity* to scale and provides additional flexibility as new experimental platforms are incorporated into the testbed. This architecture emulates the smart city concept in Fig. 1, and promotes loose coupling

[1]https://mqtt.org/mqtt-specification

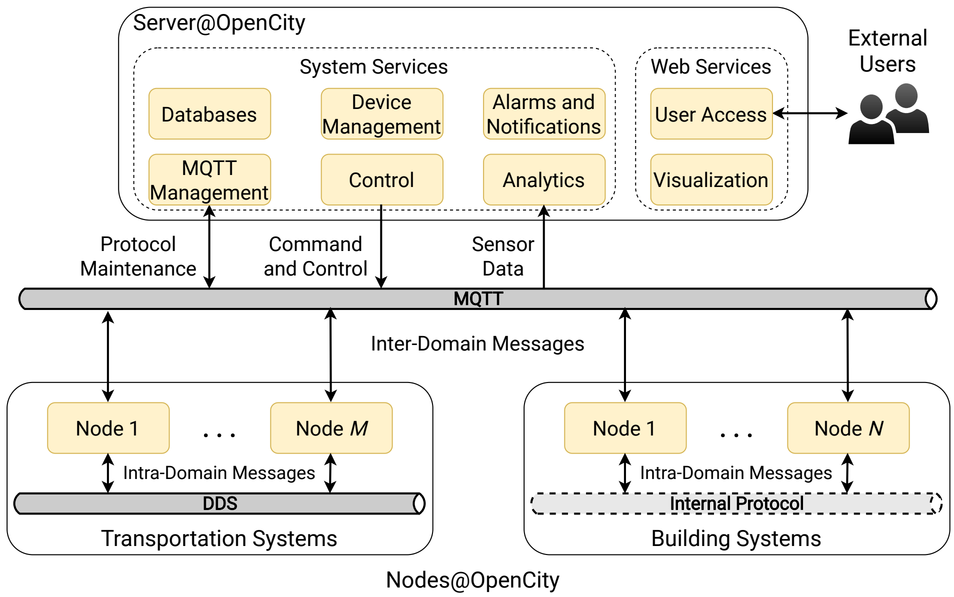


Fig. 3: *OpenCity*'s system-of-systems architecture.

among the different communication domains and testbeds using the MQTT protocol. Each main part of the architecture is discussed below.

1) Server@OpenCity: The *Server@OpenCity* hosts a web server, databases and several services that enable command, control, and sensing of the smart city platforms. The web server provides real-time visualization and a web-based user interface. *OpenCity* testbed researchers and students will use this interface to query or visualize data collected by the *OpenCity* server. Data collected from nodes are stored on the database servers with NoSQL databases. *Server@OpenCity* provides device management, control, alarm and notifications, and analytics services. It also hosts a Node-RED service as a development tool to support the connection and data processing of deployed IoT platforms within *OpenCity*.

2) Node@OpenCity: The other domains of the smart city, e.g., transportation and buildings, are managed by individual, low-power computing *nodes*, such as Raspberry Pis. Each *OpenCity* node, called *Node@OpenCity*, maintains its local data and communicates with other nodes on its chosen internal protocol. For example, the transportation system uses a DDS-based protocol to coordinate among transportation agents; however, buildings may also use MQTT for internal coordination. To formalize a common set of MQTT message payloads across *OpenCity*, we define and implement all data types with Google Protocol Buffers[2]. Using this approach, each message is formally defined and compiled into multiple implementation languages, allowing implementation flexibility within each *OpenCity* sub-domain. This architecture allows developers to choose the communication infrastructure that best suits a domain's needs.

[2]https://developers.google.com/protocol-buffers

IV. OpenCity Intelligent Transportation System

The *OpenCity* transportation system comprises devices relevant to intelligent transportation use cases: autonomous vehicles, road sensors, and traffic signals. The transportation leverages the Robot Operating System version 2 (ROS2) to maximize code re-use and ease algorithm integration into all agents within the transportation infrastructure. This framework provides modern robotics and autonomy software programming and networking abstractions. Furthermore, more recent industry platforms, e.g., AutoWare[3] and CARMA[4], use the ROS2 framework to implement their core autonomy stack.

Fig. 4 illustrates the transportation system's network architecture and its interconnection to the *OpenCity* testbed. ROS2 provides the communication fabric among transportation agents over the *OpenCity*'s WiFi network. Communication to and from the *OpenCity* core server and other non-ROS2 based infrastructure is facilitated by *bridging* the autonomous systems to *OpenCity*'s MQTT network. This network structure facilitates the management, decision support, and command services that reside on the testbed's more capable servers. The *OpenCity* intelligent transportation system design has two subsystems: vehicles and traffic signals. These intelligent agents communicate and coordinate amongst each other when running local autonomy algorithms; however, the MQTT bridging mechanism enables centralized control use cases.

The prototype intelligent transportation architecture is tested on two of our experimental platforms: a modified Waveshare PiRacer (Fig. 5a) and a Raspberry Pi based traffic signal module (Fig. 5b). Like their real-world counterparts, these platforms are equipped with a variety of sensors. The Piracer uses the Raspberry Pi Camera Module for its environment

[3]https://www.autoware.auto/
[4]https://highways.dot.gov/research/operations/CARMA-products

perception and a Raspberry Pi 4 for its computation. Fig. 5b shows one of the traffic signal prototypes for the transportation testbed. It uses a Raspberry Pi 4 as the controller and the PiTraffic hat for the LED stacks.

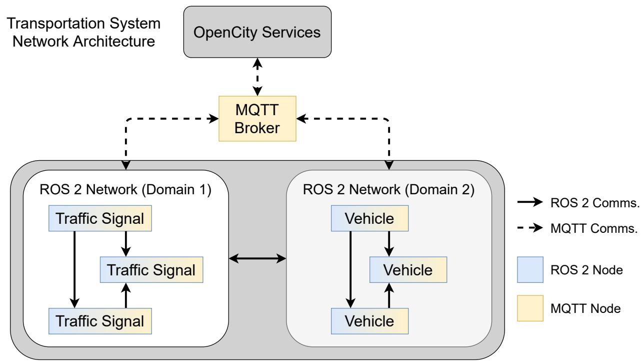
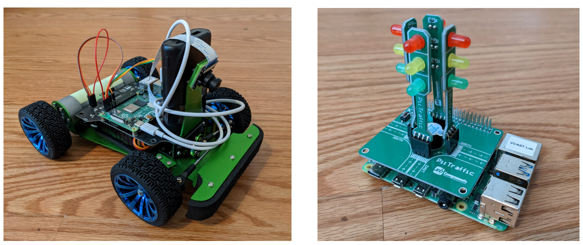


Fig. 4: High-level communication architecture for the transportation system.



(a) (b)

Fig. 5: The transportation system's hardware platforms.

V. OpenCity Interactive Building System

In the United States, residential and commercial buildings account for 40% of the total energy use, 70% of the electricity use, 36% of the total greenhouse gas emissions, and 12% of fresh water consumption. A proper building control framework can help reduce up to 30% of energy costs [15]. Moreover, since Americans spend 90% of their lives in buildings [16], it is clear that an efficient building management system can save time, money, and energy. Intelligent buildings would learn occupants' energy needs, integrate renewable energy and flexible loads into their energy forecasts, respond to changing weather conditions and uncertainty in renewable outputs [17].

While energy management products for a single home or building are available today, development of real-time, networked control of multiple buildings is still in its infancy. In the *OpenCity* testbed, smart buildings integrate various techniques and technologies to create a facility that is safer, more comfortable, efficient, and cost-effective for its occupants. This testbed would allow city planners to understand interaction among building energy consumers, building schedules, and comfort requirements.

Fig. 6 shows the internal structure of the building in the VCU *OpenCity* testbed. This modular design has removable facades that allow access to the internal components of the buildings from all sides. The building framing uses T-slot aluminum that allows for the horizontal and vertical expansion of buildings. Our current designs are between three and five floors; however, the T-slot aluminum framing can be adjusted for any number of floors. As shown in Fig. 6, layers of plywood are mounted to the aluminum with brackets at regular intervals for the floors. The pictured design features an additional T-slot aluminum piece and holes cut in the plywood for an elevator. All internal walls for the buildings will be made of plywood and are designed to be adjustable so that floor plans can be rearranged as needed.

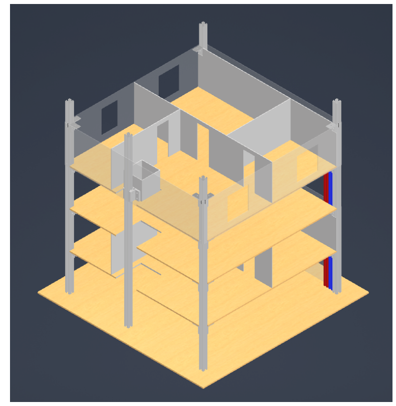


Fig. 6: Smart building internal structure.

Table I lists some of the important components used in the buildings, which are selected based on several criteria: compatibility with popular micro controllers and computers (e.g., Raspberry Pi or Arduino); low-cost; low-power; and ability to be used across multiple types of buildings. Actuators that cannot be powered off of the 5V circuit will be connected to a separate 12V circuit and operated with a Raspberry Pi controlled mechanical relay. The rest of the sensors and actuators will be wired to Raspberry Pi Zero W GPIO pins to collect data and send signals to actuators, e.g., for controlling the angle of the micro servos for opening or closing windows. Data collected on the Pi Zero Ws will be wirelessly transmitted to the Raspberry Pi 4 Model B that acts as the building manager. This building manager interfaces with the *OpenCity* MQTT communication network to send data packets to the server for storage and processing, or receive commands to pass on to the Pi Zero Ws.

Part	Usage
Raspberry Pi 4 Model B	Building Control and Management
Raspberry Pi Zero W	Sensor/Actuator Control and Management
Micro servo	Door and Window Movement
DHT22 Sensor	Temp and Humidity Monitoring
MQ-2 Gas Sensor	Flammable Gas and Smoke Sensor
Light Sensor	Light Level in Building
Passive Infrared (PIR)	Motion Sensor
IR LED	PIR Trigger / Emulate Human Presence
Raspberry Pi Camera	Building Security and Monitoring
Electric Heating Pad	Individual Room Heating
5V Cooling Fan	Small Fan for Cooling
RGB LED strip	Lighting in Buildings
Water Level Sensor	Tank Water Level and Leaks

TABLE I: Buildings component list

The following subsections describe *OpenCity*'s four distinct buildings: a) Residential building, b) Commercial office building, c) Hospital building, and d) Water treatment plant.

A. Residential and Office Buildings

The residential and office buildings will have a similar distribution of sensors and actuators per room. Sensors monitor temperature and humidity, light levels, motion (which will be triggered with an IR LED to emulate human presence), and smoke and flammable gasses. These sensors can also monitor other comfort factors such as noise level as needed. Actuators will be included in both building types to control physical features such as doors and windows.

B. Hospital Building

The hospital building will have the same sensors and servos as the residential and office buildings. Hospital buildings have designated bathrooms, waiting, diagnostic, and triage rooms. This building integrate with VCU's Medical Device Security lab. It will be possible to query data through this building and receive data collected from real medical devices. The Medical Device Security lab has three focal points in working to improve security in commercial and home healthcare environments: Offensive security and penetration testing, FPGA based secure by design systems, and secure IoT and sensing applications.

All digital network communication between the *OpenCity* lab and Medical Device Security lab will occur through the Medical Device Security Lab firewall using RSA-4096 public/private key authentication. Inbound and outbound connections between the labs will only be allowed to and from the dedicated *OpenCity* IoT VLAN, which will be for sensing and IoT devices that communicate with *OpenCity*'s hospital building. This structure ensures the safety of both the *OpenCity* and Medical Device Security labs and the integrity of the penetration testing and development environments within the Medical Device Security Lab.

IoT devices and sensors may live within the *OpenCity* hospital building or within the Medical Device Security Lab. Linux based virtual machines or devices communicate between the two labs using keyed authentication. It would be advantageous to put industrial grade and realistic size sensors in the medical device security lab on the IoT network. An example experiment may be a sensor placed in hospital beds that reports usage data back to the smart city for purposes, such as indicating hospital capacity to emergency services. The lab could also house standard building sensors that feed back into the smart city lab.

Sensor data collected within the Medical Device Security Lab and output from medical devices will be used to build digital simulations of this equipment within the *OpenCity* hospital building. These simulations can be updated as new device behavior is discovered, to include vulnerabilities found within the devices themselves where applicable to the Smart City Research. Examples of existing equipment in the Medical Device security lab are: Connected ultrasounds, patient monitors, ventilators, and IV pumps.

C. Water Distribution System

The water distribution system supplies faucet water to the three key buildings in the smart city: the residential building, hospital, and business building. Fig. 7 is a simplified process flow diagram for the system, showing only the residential building details. The water distribution system comprises a supply tank, a Variable Frequency Drive (VFD) driven pump, water distribution piping, and control valves that represent user consumption. The consumed water is directed to a single return tank that serves the three buildings. The return tank is equipped with an on/off pump that circulates the returned water to the three buildings' water supply tanks, comprising a closed-loop circulation system.

The system is equipped with the necessary instrumentation to measure tank levels, pump discharge pressures, and the flow rate at each floor. The speed control of each supply tank pump enables the control of the hydraulic system operating point (pressure, flow) under a variety of user consumption profiles. The testbed will be used to assess and compare the performance of several control techniques, including classical feedback control, feed-forward control, and reinforcement learning.

The system is designed to allow implementing and testing both centralized and distributed control designs. Each actuator, e.g., a pump or a valve, has its own embedded controller that has local wireless communication with the relevant sensors. This provides more flexibility in adopting advanced control techniques, e.g., feed-forward control by reading user consumption valve position sensors. To safeguard against system safety hazards, e.g., over-pressure, an independent safety system is utilized with its own sensors to provide an additional layer of protection. The overall system is connected to the smart city cyber infrastructure via MQTT broker communication.

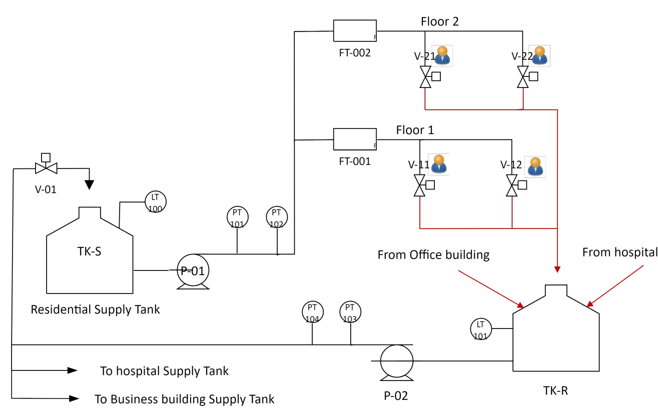


Fig. 7: A simplified process flow diagram for the water distribution system. Details for the office and hospital buildings are the same as the residential building and omitted to avoid cluttering the diagram.

The key performance measures for the system are the energy consumed in water distribution, and a reliable water distribution. The ultimate objective is to minimize the energy and water consumption while reliably satisfying the user convenience constraints, e.g., minimum flow rate. To enable system performance evaluation, we develop a stochastic model for the user consumption profile, which includes the time of the day, the duration of opening each valve, and the percentage opening (water flow) for each usage. The model accounts for different genders and age groups, and differentiates between the consumption of different building types. These models enable us to run stochastic simulations to assess the performance of different control strategies under a variety of user consumption patterns.

VI. OpenCity Management Architecture

OpenCity management architecture is developed to ensure four testbed performance objectives: cost-effectiveness, users' comfort, safety, and security. Fig. 8 presents an overview of *OpenCity* management architecture. The architecture consists of three main blocks: the system module, environment module, and control module. In the system module, the dynamics of *OpenCity* components are modeled and tuned through model-based forecasting strategies or learning. In the environment module, the environment variations are predicted based on the offline historical data and online measurements. The current and future values of system dynamics are streamed into the control module, where a cost function is formulated with desired performance specifications, such as safety, cost-effectiveness, and convenience. Optimal control decisions are generated by solving the optimization problem in the control module, and they are fed into the testbed actuators.

From the modeling viewpoint, two types of management strategies are deployed depending on the component specifications and requirements; i.e., model-based and learning-based management. Model-based techniques are implemented for controlling *OpenCity* processes with well-understood behavior, such as indoor condition regulation in smart buildings. In a model-based design, a mathematical representation of process dynamics is utilized to minimize the deviation of the controlled variables from the desired values [18]. Although model-based control approaches are a well-studied framework, creating an exact mathematical model for the complex processes is a key challenge [19]. Also, *OpenCity* components are subject to uncertainties and modeling such systems is difficult [20].

Learning-based management techniques are implemented for *OpenCity* components that are difficult to model from first principles, such as occupants' perception of comfort/feedback, and environmental disturbances. Since a learning-based management system does not utilize models to describe component characteristics, its performance is not affected by modeling inaccuracies. In the management architecture, learning-based management structures are exploited along with the model-based management techniques to take into account the dynamic information of users and environmental disturbances during *OpenCity* decision-making process. Integrating this information into the management system leads to a design that enables adaptation to uncertain variations and real-time performance improvement over time.

OpenCity components operate in uncertain physical environment; therefore, decision making for the system is strongly influenced by unexpected disruptive events. One of the important challenges in *OpenCity* management is ensuring its reliability and adaptability to unforeseen crises. To enhance the decision-making process in the event of these situations, we develop a decision support system (DSS) that operates along with the management system. The decision support system uses Markov decision processes (MDP) to model a wide range of common disruptive scenarios that may occur in the *OpenCity* management system. In the event of a critical situation, an initial action is chosen from a set of possible actions, and a reward value is assigned to it. Then, by transiting to new states and adopting actions with higher reward values, the decision making process is gradually improved. The developed *OpenCity* DSS anticipates, adapts, and rapidly recovers from unexpected emergencies.

VII. Conclusion

The smart city concept promises citizens improved quality of life through increased access to public services. To do so, smart cities require a scalable architecture that integrates heterogeneous sensing, communication, control, and analytics technologies. There are multiple application-level protocols that enable smart city data collection, e.g., MQTT, CoAP (Constrained Application Protocol), WebSocket, and analytics, e.g., Apache Kafka, Amazon Kinesis. However, there is still the need for an open architectures and secure platform to enable researchers, application developers, and city planners to easily discover, assess, and mitigate real-life smart city challenges and problems. The presented smart city testbed, *OpenCity*, meets these requirements through its data collection and processing units, database management, distributed performance management algorithms, and real-time data visualization.

The main contribution of the *OpenCity* is to provide an accessible experimental environment for researchers to explore algorithms and technologies that would be integrated into intelligent, sustainable cities. The *OpenCity* testbed includes four types of smart buildings (including residential, commercial office, hospital, and water treatment plant), intersecting roads, intelligent transportation systems (including autonomous vehicles and traffic signals), as well as the associated communication backbone and data analytics infrastructure. In the testbed architecture, *Server@OpenCity* serves as a centralized data hub and access that provides several software services to platform users. The intelligent nodes, *Nodes@OpenCity*, maintain their local data, communicate with other nodes and *Server@OpenCity*. In addition, the integrated management systems (i.e., model-based predictive control approaches and learning-based techniques) enable real-time management of the *OpenCity* subsystems considering various uncertainties, including user behavior and environmental variations.

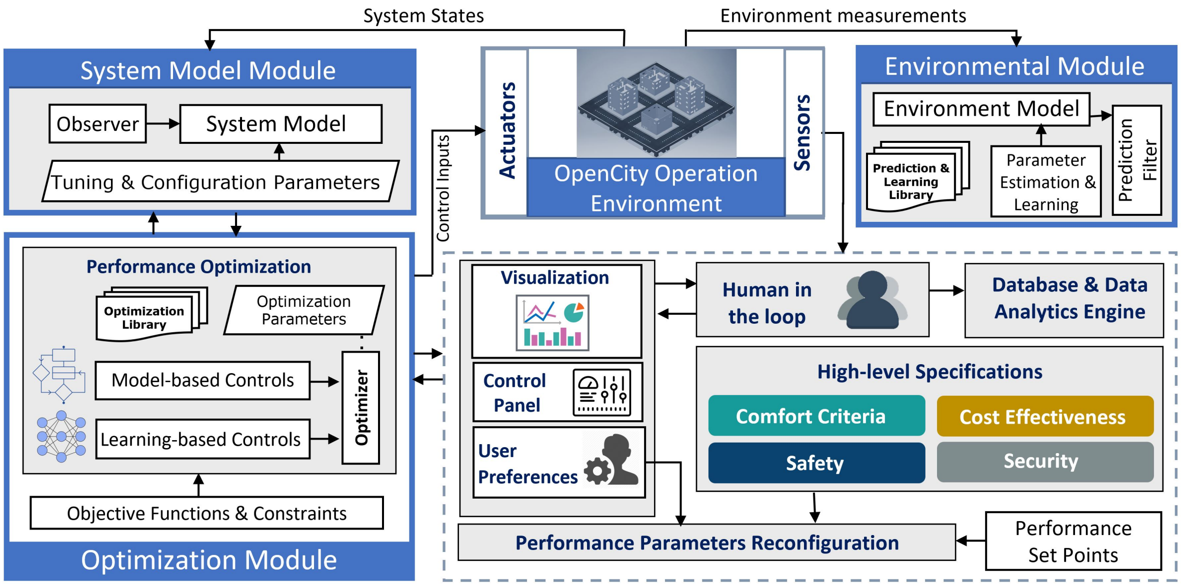


Fig. 8: *OpenCity* management architecture.

The design and development of the *OpenCity* testbed contributes to both the scientific community and society by (a) developing novel collaborative and interdisciplinary research approaches (b) training students in advanced topics and technologies for addressing real-life challenges that our community faces, (c) providing guidance and support to local communities to employ advanced technologies and innovative management systems in their smart city plans, and (d) increasing partnerships between academia and industry. Through the presented architecture with software services, it is expected that *OpenCity* can support various educational and research activities related to smart city development.

Acknowledgment

This work is supported by the Commonwealth Cyber Initiative (CCI), an investment in the advancement of cyber R&D, innovation, and workforce development in Virginia. For more information about CCI, visit cyberinitiative.org.

References

[1] C. G. Cassandras, "Smart cities as cyber-physical social systems," *Engineering*, vol. 2, no. 2, pp. 156–158, 2016.
[2] United Nations (2015), "United nations department of economic and social affairs, population division, world urbanization prospects: The 2014 revision," 2015.
[3] B. N. Silva, M. Khan, and K. Han, "Towards sustainable smart cities: A review of trends, architectures, components, and open challenges in smart cities," *Sustainable Cities and Society*, vol. 38, pp. 697–713, 2018.
[4] "A strategy for american innovation," *National Economic Council and Office of Science and Technology Policy*, 2015.
[5] B. Dong, V. Prakash, F. Feng, and Z. O'Neill, "A review of smart building sensing system for better indoor environment control," *Energy and Buildings*, vol. 199, pp. 29–46, 2019.
[6] A. Nugur, M. Pipattanasomporn, M. Kuzlu, and S. Rahman, "Design and development of an iot gateway for smart building applications," *IEEE Internet of Things Journal*, 2018.
[7] Y. Li, X. Cheng, Y. Cao, D. Wang, and L. Yang, "Smart choice for the smart grid: Narrowband internet of things (NB-IoT)," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1505–1515, 2017.
[8] M. Veres and M. Moussa, "Deep learning for intelligent transportation systems: A survey of emerging trends," *IEEE Transactions on Intelligent transportation systems*, 2019.
[9] J. Karwot, J. Kaźmierczak, R. Wyczółkowski, W. Paszkowski, and P. Przystałka, "Smart water in smart city: A case stud," in *Proceedings of SGEM 16th International Scientific Conference on EARTH&GEOSCIENCES, book*, vol. 3, 2016, pp. 851–858.
[10] M. I. Pramanik, R. Y. Lau, H. Demirkan, and M. A. K. Azad, "Smart health: Big data enabled health paradigm within smart cities," *Expert Systems with Applications*, vol. 87, pp. 370–383, 2017.
[11] A. S. Elmaghraby and M. M. Losavio, "Cyber security challenges in smart cities: Safety, security and privacy," *Journal of advanced research*, vol. 5, no. 4, pp. 491–497, 2014.
[12] 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," *Computer Networks*, vol. 61, pp. 217–238, 2014, special issue on Future Internet Testbeds – Part I.
[13] C. Adjih, E. Baccelli, E. Fleury, G. Harter, N. Mitton, T. Noel, R. Pissard-Gibollet, F. Saint-Marcel, G. Schreiner, J. Vandaele, and T. Watteyne, "FIT IoT-LAB: A large scale open experimental IoT testbed," in *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, 2015, pp. 459–464.
[14] S. Latre, P. Leroux, T. Coenen, B. Braem, P. Ballon, and P. Demeester, "City of things: An integrated and multi-technology testbed for IoT smart city experiments," in *2016 IEEE International Smart Cities Conference (ISC2)*, 2016, pp. 1–8.
[15] L. Pérez-Lombard, J. Ortiz, and C. Pout, "A review on buildings energy consumption information," *Energy and buildings*, vol. 40, no. 3, pp. 394–398, 2008.
[16] G. W. Evans and J. M. McCoy, "When buildings don't work: The role of architecture in human health," *Journal of Environmental psychology*, vol. 18, no. 1, pp. 85–94, 1998.
[17] P. Geraldo Filho, L. A. Villas, V. P. Gonçalves, G. Pessin, A. A. Loureiro, and J. Ueyama, "Energy-efficient smart home systems: Infrastructure and decision-making process," *Internet of Things*, vol. 5, pp. 153–167, 2019.
[18] E. F. Camacho and C. B. Alba, *Model predictive control*. Springer Science & Business Media, 2013.
[19] M. Benosman, *Learning-based adaptive control: An extremum seeking approach–theory and applications*. Butterworth-Heinemann, 2016.
[20] J. Kleissl and Y. Agarwal, "Cyber-physical energy systems: Focus on smart buildings," in *Design Automation Conference*. IEEE, 2010, pp. 749–754.