International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 13 Issue: 06 | Jun 2026 www.irjet.net p-ISSN: 2395-0072

# Agent-based Software Technology and Cyber security in Smart Cities

**Abdulmalik Mubarak Al-Harbi**

*College of Computing, Fahad Bin Sultan University, AIMD Lab, KSA, Tabuk*

---

**Abstract -** *This paper studies agent-based software technology and cyber security in smart cities from a software architecture perspective. Agent-based systems model software components as autonomous or semi-autonomous agents that perceive their environment, make decisions, communicate and act toward defined goals. This paradigm is valuable for distributed environments such as smart cities, where transport, energy, emergency services, sensors and citizen platforms interact continuously. The paper explains the concepts of agents, multi-agent systems, autonomy, reactivity, proactiveness, communication and coordination. It then analyses smart-city cyber security, including attack surfaces, IoT risk, data integrity, monitoring, incident response and operational resilience. The paper argues that agent-based systems can improve scalability and local responsiveness, including in cyber security monitoring, but they require strong governance, authenticated communication, auditability and human oversight. The conclusion emphasizes secure autonomy: smart-city agents should be powerful enough to support adaptive services but constrained enough to remain safe, transparent and accountable.*

**Key Words: Agent-based Software Technology, Cyber security in Smart Cities, governance, risk management, digital systems.**

## 1. INTRODUCTION

Modern software systems increasingly operate in distributed, dynamic and data-rich environments. Traditional centralized software can be difficult to adapt when many components must make local decisions, cooperate with other components and respond to changing conditions. Agent-based software technology addresses this challenge by modelling software entities as agents that can perceive, decide, act and communicate [1]. This approach is especially relevant to smart cities, where transport, energy, emergency response and public services involve many interacting subsystems.

This paper examines Agent-based Software Technology and Cyber security in Smart Cities. This paper emphasizes software architecture, autonomous agents, multi-agent coordination and defensive smart-city monitoring. It studies how agent-based systems can support distributed decision-making and how cyber security must be integrated when such agents interact with critical urban infrastructure.

The paper is organized around two questions. First, what are the concepts, applications, benefits and challenges of agent-based software technology? Second, how can cyber security be managed in smart-city software environments where many autonomous or semi-autonomous components interact? The paper argues that agent-based technology can improve adaptability and scalability, but only if communication, coordination, verification and security controls are carefully designed.

## 2. AGENT-BASED SOFTWARE TECHNOLOGY - BACKGROUND AND CONCEPTS

An agent is commonly understood as an autonomous software entity that observes its environment and acts to achieve goals. Multi-agent systems involve several agents interacting with each other through communication, coordination, negotiation or competition. The agent paradigm is useful when a system is distributed, open, dynamic or too complex for a single centralized controller. Classic research by Wooldridge and Jennings helped establish intelligent agents as a major concept in distributed artificial intelligence and software engineering. (Wooldridge & Jennings, 1995; Wooldridge, 1997) [2]. Figure 1 below illustrates the Hierarchical Multi-Agent.

Figure 1 illustrates a multi-agent chatbot architecture for a city information system. The user interacts with a chatbot interface, where the user submits a query and receives a system-generated answer. The chatbot is connected to the city information system, which supports the retrieval and delivery of relevant city-related information. At the center of the architecture is the agent-conductor, which acts as the main coordinator. It receives the user request and manages communication among different specialized agents. The system includes secondary agents, such as an agent for tool calling, an agent for RAG operation, and an agent for answer summarization. These agents interact with external resources, including the city services API and a vector database with documents, to retrieve accurate and context-aware information. Figure 1 also includes agent-validator components, which help check or validate the outputs before they are returned to the user. The arrows show two types of interactions: external communication among agents and tool usage. Overall, the figure represents a coordinated AI-based system where multiple agents work together to process user queries, retrieve city information, validate results, and generate a summarized final response.

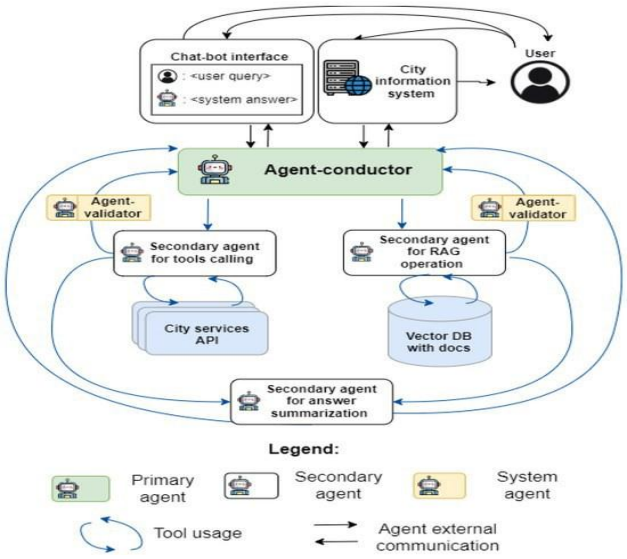


**Figure-1:** Hierarchical Multi-Agent.

Agent-based software differs from ordinary object-oriented software because agents are usually described in terms of autonomy, reactivity, proactiveness and social ability. Autonomy means the agent can operate without constant human or central control. Reactivity means the agent responds to environmental changes. Proactiveness means the agent can pursue goals. Social ability means the agent can communicate or coordinate with other agents.

A multi-agent architecture usually includes an environment, agent layer, communication mechanism, coordination protocol, service layer and governance controls. Agents may be rule-based, belief-desire-intention based, learning-based or hybrid [3]. Their internal design depends on the application domain. A warehouse scheduling system may use negotiation agents, while a traffic system may use agents that coordinate signals, detect congestion and communicate with emergency-service modules.

Agent communication is a key concept. Standards such as the FIPA Agent Communication Language were developed to support interoperability between agents. In practice, modern implementations may use APIs, message queues, event buses or semantic communication protocols [4]. The important point is that agents need structured communication; otherwise, distributed autonomy becomes unmanageable.

## 3. SOFTWARE AGENTS APPLICATIONS, BENEFITS, AND CHALLENGES

Agent-based software technology can be applied in simulation, logistics, smart grids, traffic management, e-commerce, robotics, cloud services and cyber security. In simulation, agents can represent people, vehicles, devices or organizations and help researchers study complex behaviour. In logistics, agents can allocate tasks, negotiate schedules and adapt routes. In smart grids, agents can support distributed energy management. In cyber security, agents can monitor local events and cooperate to detect anomalies. There is a strong link between location based services and systems built using agent software technology. The key idea of location based services is to enable users to search for nearest points of interests, such as nearest hospitals, sport clubs, or universities [5, 6, 7, 8]. Agents based systems contribute to achieve load balancing between privacy protection and power consumption of mobile devices of users [9], and also to enhance privacy protection in location based services when users are searching for moving points of interests [10].

The main benefits are modularity, scalability, adaptability and resilience, especially when it comes to talking about big data and application of related techniques such as data mining technique [11]. A multi-agent system can distribute decision-making across components, allowing local agents to respond quickly while still sharing information with the wider system. This can reduce the burden on a central controller and support systems that evolve over time. Agent-based design also maps naturally to real-world systems made of many semi-independent actors. This benefit can be highlighted in medical sectors, where intelligent diagnostic systems, such as those proposed in [12, 13, 14], when such systems are enhanced by integrating with agents.

However, agent-based systems are not simple to engineer. Coordination can be difficult because agents may have incomplete information or conflicting goals. Testing is challenging because system behaviour emerges from interactions rather than from a single execution path. Verification, debugging and accountability become harder when agents learn or adapt. Security is also critical, especially in web applications [15], because compromised agents may spread false information or manipulate decisions. Actually, authors of work [16] presented a comprehensive survey related to security of agents and the related attacks. There are some works related to enhance security of agents, such as [17, 18], where the dummy based and self-protection approaches are employed.

For these reasons, agent-based software engineering requires disciplined design. Developers should define agent responsibilities, communication protocols, trust boundaries, logging mechanisms and failure handling. The architecture should also include governance controls so that autonomous behaviour remains observable and auditable.

## 4. CYBERSECURITY IN SMART CITIES - BACKGROUND AND CONCEPTS

Smart-city cybersecurity protects connected urban software and infrastructure from unauthorized access, disruption, manipulation and data exposure. From a software architecture perspective, smart cities are systems of systems. Traffic controllers, CCTV platforms, environmental sensors, citizen apps, emergency systems and cloud analytics may all exchange data. This creates a complex attack surface that cannot be secured only by perimeter defences.

The NIST Cybersecurity Framework 2.0 [19] provides a useful structure because it includes governance as a core function alongside identifying, protecting, detecting, responding and recovering. In agent-supported smart cities, these functions can be mapped to software responsibilities. Agents can help identify anomalies, detect device failures, prioritize alerts and coordinate response workflows. However, the agents themselves must be secured.

CISA and partner agencies highlight smart-city risks such as expanded attack surfaces, interconnected infrastructure, supply-chain exposure and operational technology integration [20]. These risks are important because smart-city services may affect physical systems. A manipulated traffic signal, water sensor or emergency communication system can create consequences beyond information loss. Cybersecurity is therefore connected to public safety and service continuity.

Cybersecurity in smart cities also requires attention to data integrity. Many smart-city decisions depend on sensor readings and event streams. If an attacker falsifies or manipulates data, the system may make wrong decisions while appearing operational. Integrity checks, authentication, anomaly detection and redundancy are therefore necessary.

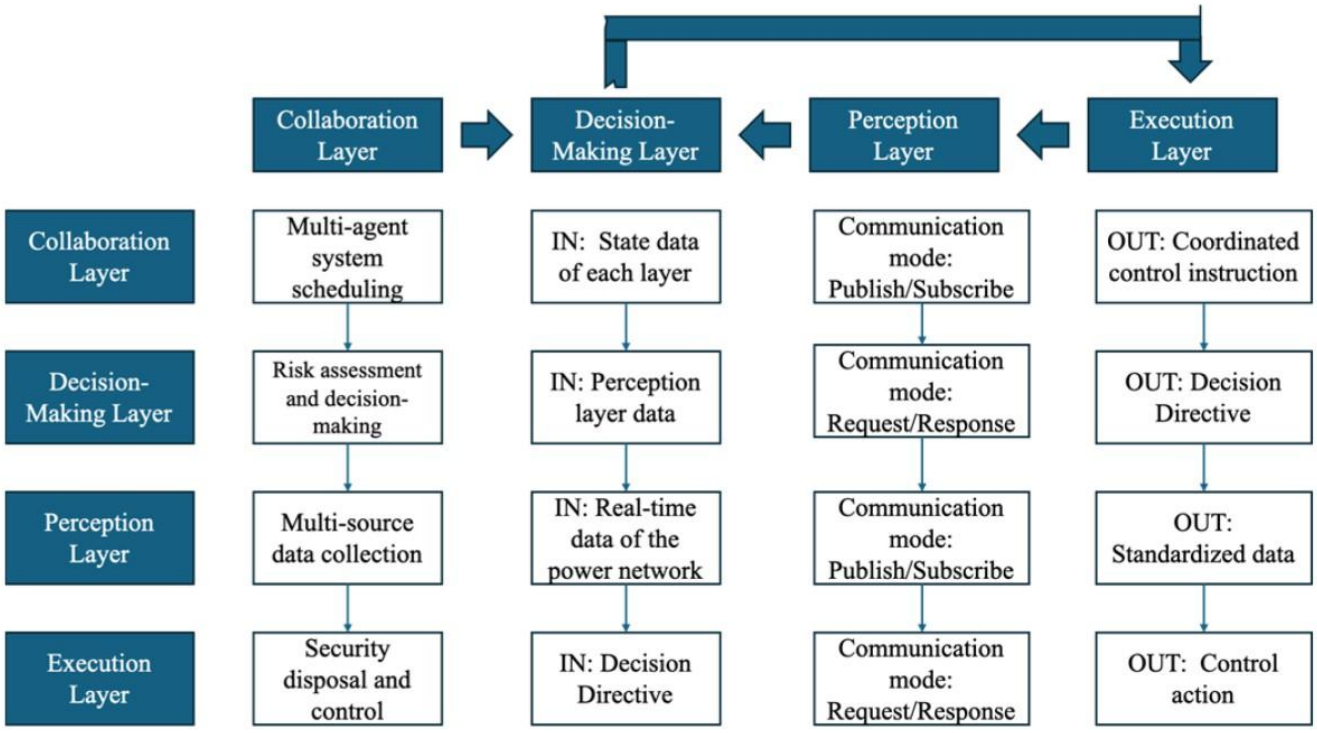


**Figure -2:** Cyber security-Oriented Agent Hierarchy for Smart Grid Protection

Figure 2 presents a layered multi-agent control architecture for a power network or smart infrastructure system. It is organized into four main layers: collaboration layer, decision-making layer, perception layer, and execution layer. The collaboration layer handles multi-agent system scheduling and coordination among agents. The decision-making layer receives state and perception data, performs risk assessment, and generates decision directives. The perception layer collects real-time data from the power network and manages communication

Using modes such as publish/subscribe and request/response. Finally, the execution layer transforms decisions into coordinated control instructions, standardized data, and practical control actions. Therefore, the figure shows how information flows from data collection and perception toward decision-making and execution, while feedback and communication between layers ensure coordinated, secure, and adaptive system control.

## 5. Cyber security APPLICATIONS, BENEFITS, AND CHALLENGES

Cyber security applications in smart cities include device monitoring, network intrusion detection, vulnerability management, incident response, physical asset protection and resilience planning. Agent-based approaches can contribute by distributing monitoring across neighborhoods, infrastructure sectors or data streams. For example, a local agent may observe unusual traffic-signal behavior and send an alert to a coordination agent that compares it with other events.

The benefit of agent-supported cyber security is faster local awareness combined with wider situational understanding. Instead of sending every raw event to a central platform, local agents can filter, classify and summarize events. Coordination agents can then correlate signals across sectors. This supports scalability and reduces the chance that important signals are lost in large volumes of data.

The challenges include trust management, false alerts, adversarial manipulation, secure communication and operational accountability. If a monitoring agent is compromised, it may hide incidents or generate misleading alerts. If agents use weak authentication, attackers may impersonate them. If automated response is poorly designed, it may disrupt legitimate services. Therefore, agent-based cybersecurity must be designed with strong identity, authorization, logging and human oversight.

Smart-city cybersecurity also faces institutional challenges. Multiple departments and vendors may control different systems, and not all have the same security maturity. Technical tools alone cannot solve coordination problems. Governance, contracts, training and incident exercises are needed to make cybersecurity operational.

Figure 3 shows a multi-agent learning and classification framework for processing smart city IoT data. It begins with different smart city devices and sensors that collect raw IoT data. This data is first handled by a loading agent, which prepares the initial traffic dataset for further processing. Next, the data passes through a data preprocessing agent, supported by preprocessing subagents, to clean and organize the dataset. The processed data is then sent to a partitioning clustering agent, which groups or partitions the data before classification. After that, a lazy classification agent, supported by classification subagents, produces a labeled dataset. The labeled dataset is passed to a cost-sensitive meta-classification agent, which works with an expert system agent to adjust learning parameters and identify the optimum model configuration. Finally, the optimized dataset and model configuration are delivered to the output agent. Therefore, the figure illustrates how multiple specialized agents cooperate to transform raw smart city sensor data into classified and optimized decision-support outputs.

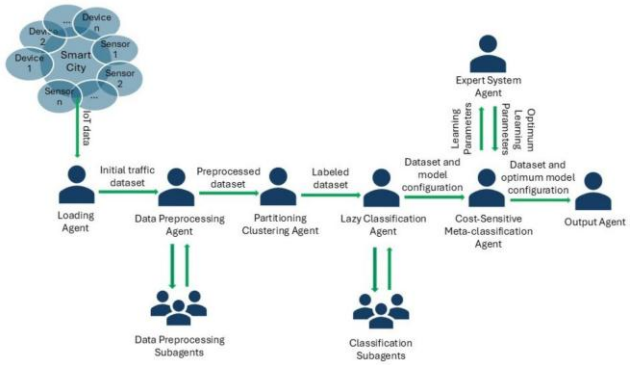


**Figure-3:** Real-Time Multi-Agent System for IoT Intrusion Detection

## 6. COMPARATIVE DISCUSSION

Agent-based software technology and smart-city cybersecurity intersect in two ways. First, agent-based systems can be used to build smart-city services, such as adaptive traffic control, energy optimization and emergency coordination. Second, agents can be used as cybersecurity components that monitor, detect and support response. In both cases, the design must account for autonomy, trust and accountability.

The similarity between the topics is their reliance on distributed decision-making. Smart cities cannot be managed efficiently through isolated manual processes, and multi-agent software provides a way to coordinate complex subsystems. The difference is that agent-based technology is a software design paradigm, while cybersecurity is a risk management and protection discipline. When combined, they create powerful but sensitive systems.

A useful principle is secure autonomy. Agents should be autonomous enough to respond locally, but constrained enough to remain safe, observable and accountable. This requires clear policies, authenticated communication, access control, audit trails and escalation to human operators for high-risk decisions.

## 7. ETHICAL, LEGAL, AND PROFESSIONAL CONSIDERATIONS

Ethical issues include transparency, accountability and the limits of automation. If agents make decisions affecting public services, citizens and operators should know how responsibility is assigned. Automated decisions in traffic, public safety or resource allocation can have unequal impacts if the design ignores social context. Agent-based systems should therefore be evaluated for fairness and human oversight, not only efficiency.

Legal considerations include cybersecurity obligations, procurement rules, data protection, incident papering and liability. Smart-city systems may be operated by vendors, municipal departments and regional authorities, which

complicates responsibility. Contracts should specify security requirements, update obligations, audit rights and incident notification duties. Autonomous agents should not create gaps where no party is clearly responsible.

Professional responsibility requires developers to test multi-agent behaviour under normal, failure and adversarial conditions. Security teams should threat-model agent communication, validate logs and prepare response plans. Public managers should ensure that automation serves public value rather than replacing accountability.

## 8. RECOMMENDATIONS

Organizations developing agent-based smart-city systems should begin with a clear architectural model. Each agent should have a defined role, data access level, communication method and authority boundary. Critical decisions should include human oversight or approval. Developers should also design for observability so that agent actions can be audited after incidents.

Security controls should include authenticated agent communication, encryption, least-privilege access, secure update mechanisms, tamper-resistant logging and monitoring for abnormal behaviour. Agents should not blindly trust messages from other agents. Trust should be established through identity, authorization and validation of data quality.

Smart-city authorities should run tabletop exercises and simulations that include agent failure, false data injection and vendor outage scenarios. They should also require documentation from suppliers about security design, support periods and vulnerability disclosure. Finally, agent-based systems should be introduced gradually, beginning with decision-support roles before moving toward higher levels of automation.

## 9. CONCLUSION

Agent-based software technology offers a powerful approach for distributed and adaptive systems. It is especially relevant to smart cities because urban services involve many interacting components, local decisions and changing conditions. Agents can support traffic optimization, energy management, monitoring and cybersecurity operations. However, the same autonomy that makes agents useful also creates risks related to coordination, verification and accountability.

Cybersecurity in smart cities must therefore be integrated into agent-based design from the beginning. Secure autonomy, strong communication controls, logging, governance and human oversight are essential. The most effective future smart-city systems will not be the most automated systems, but the systems that combine adaptive software with safety, resilience and public accountability.

## ACKNOWLEDGEMENT

ChatGPT was used only to support the preparation of illustrative figures and the proofreading of the manuscript, while the main scientific ideas, analysis, and content were developed and presented by the author.

## REFERENCES

[1] Abar, Sameera, et al. "Agent Based Modelling and Simulation tools: A review of the state-of-art software." Computer Science Review 24 (2017): 13-33.

[2] Wooldridge, M. J. "Agents Theories, Architectures and Languages: A Survey. In Intelligent Agents": ECAI-94 Workshop on ATAL, eds Wooldridge and Jennings." (1995): 1-39.

[3] Dorri, Ali, Salil S. Kanhere, and Raja Jurdak. "Multi-agent systems: A survey." Ieee Access 6 (2018): 28573-28593.

[4] Poslad, Stefan, Phil Buckle, and Rob Hadingham. "The FIPA-OS agent platform: Open source for open standards." Proceedings of PAAM. Vol. 2000. 2000.

[5] Mohamad Shady Alrahhal, Muhammad Usman Ashraf, Adnan Abesen and Sabah Arif, "AES-Route Server Model for Location Based Services in Road Networks," International Journal of Advanced Computer Science and Applications, vol. 8, no. 8, 2017.

[6] Hosam Alrahhal et al., "A symbiotic relationship based leader approach for privacy protection in location based services," ISPRS International Journal of Geo-Information, vol. 9, no. 6, p. 408, 2020.

[7] Mohamad Shady Alrahhal, Maher Khemakhem and Kamal Jambi, "A Survey on Privacy of Location-Based Services: Classification, Inference Attacks, and Challenges," Journal of Theoretical and Applied Information Technology, vol. 95, no. 24, 2017.

[8] Abdullah S. Alyousef, Karthik Srinivasan, Mohamad Shady Alrahhal, Majdah Alshammari and Mousa Al-Akhras, "Preserving Location Privacy in the IoT against Advanced Attacks using Deep Learning," International Journal of Advanced Computer Science and Applications, vol. 13, no. 1, 2022.

[9] Mohamad Shady Alrahhal, Maher Khemekhem and Kamal Jambi, "Achieving load balancing between privacy protection level and power consumption in location based services," International Research

Journal of Engineering and Technology, vol. 5, no. 3, pp. 619-625, 2018.

[10] Mohamad Shady Alrahhal, Maher Khemakhem and Kamal Jambi, "Agent-Based System for Efficient kNN Query Processing with Comprehensive Privacy Protection," International Journal of Advanced Computer Science and Applications, vol. 9, no. 1, 2018.

[11] Mohamad Shady Alrahhal and Adnan Abi Sen, "Data mining, big data, and artificial intelligence: An overview, challenges, and research questions," 2018.

[12] Mona Alfifi, Mohamad Shady Alrahhal, Samir Bataineh and Mohammad Mezher, "Enhanced Artificial Intelligence System for Diagnosing and Predicting Breast Cancer using Deep Learning," International Journal of Advanced Computer Science and Applications, vol. 11, no. 7, 2020.

[13] Mohamad Shady Alrahhal and Eftkhar Alqhtani, "Deep learning-based system for detection of lung cancer using fusion of features," International Journal of Computer Science and Mobile Computing, vol. 10, no. 2, pp. 57-67, 2021.

[14] Mohamad Shady Alrahhal and Majed Abdullah Albarrk, "A Survey of the COVID-19 Epidemic Through the Eyes of Artificial Intelligence and Deep Learning: Challenges and Research Questions," 2020.

[15] Majed Abdullah Albarrk and Mohamad Shady Alrahhal, "Web Applications Security: More Collaboration," 2020.

[16] Bandar Alluhaybi, Mohamad Shady Alrahhal, Ahmed Alzhrani and Vijey Thayananthan, "A Survey: Agent-based Software Technology Under the Eyes of Cyber Security, Security Controls, Attacks and Challenges," International Journal of Advanced Computer Science and Applications, vol. 10, no. 8, 2019.

[17] Bandar Alluhaybi, Mohamad Shady Alrahhal, Ahmed Alzhrani and Vijey Thayananthan, "Dummy-based approach for protecting mobile agents against malicious destination machines," IEEE Access, vol. 8, pp. 129320-129337, 2020.

[18] Bandar Alluhaybi, Mohamad Shady Alrahhal, Ahmed Alzhrani and Vijey Thayananthan, "Achieving self-protection and self-communication features for security of agent-based systems," 2020.

[19] Pascoe, Cherilyn E. "Public draft: The NIST cybersecurity framework 2.0." National Institute of Standards and Technology (2023).

[20] AlArfaj, Lulu, and Abdullah AlShuaibi. "Critical infrastructure protection." Intelligent and Secure Solutions for Digital Transformation. CRC Press 107-130.