Towards a Foundation for a Collaborative Replicable Smart Cities IoT Architecture

Alexander Nelson*
Montgomery County, MD
29 Courthouse Sq.
Rockville, Maryland 21228
Alexander.Nelson@
montgomerycountymd.gov

Greg Toth†
IoT Dev Labs
8300 Boone Blvd, Suite 500
Tysons Corner, Virginia 22182
greg@iotdevlabs.co

Daniel Hoffman
Montgomery County, MD
29 Courthouse Sq.
Rockville, Maryland 21228
Daniel.Hoffman@
montgomerycountymd.gov

Cuong Nguyen
National Institute of Standards and
Technology (NIST)
100 Bureau Drive
Gaithersburg, Maryland 20899
cuong.nguyen@nist.gov

Sokwoo Rhee
National Institute of Standards and
Technology (NIST)
100 Bureau Drive
Gaithersburg, Maryland 20899
sokwoo.rhee@nist.gov

ABSTRACT

The pervasive instrumentation of the physical world with sensors and actuators provides an unprecedented level of information granularity that is useful in decision-making processes. As municipalities and the public sector at large begin to leverage the Internet of Things (IoT) for civic solutions, there exist greater necessity and impetus to maintain a certain level of standardization in the platform and data architecture. Ideally, these standards should be in place well ahead of legislation which encourages adoption. For this reason, it is important for technologists and public policy experts to collaborate in the codification process. Currently existing Smart City deployments serve as case studies, but the unification process of best practices will require in-situ deployment and testing. This work introduces an initial investigation into a collaborative, replicable, Smart Cities, IoT Architecture through targeted real-world deployment. This paper performs the following tasks in order to actualize the envisioned collaborative framework (1) delineation of the guiding principles of the Smart-City framework, (2) development of foundational assets in an existing cloud provider, (3) demonstration of the existing functionality through a real-world use-case, and (4) call for collaboration through our online repository.

*Equally Contributing First Author
†Equally Contributing First Author

CCS CONCEPTS

•**Networks** →**Cyber-physical networks;** *Network structure; Logical / virtual topologies; Network manageability;* •**Computer systems organization** →*Distributed architectures; Dependable and fault-tolerant systems and networks;*

KEYWORDS

Smart City, Internet of Things, Cyberphysical Systems,

1 INTRODUCTION

The "Smart City" moniker, which has been adopted by many municipalities, seems to incorporate many different technologies at vastly varying scopes. The foundation of Smart City is based on the application of information and communication technology (ICT) to provide community services faster, more efficiently, or more broadly. The adoption of ICT is becoming pervasive in municipal governments. Implementations, however, can often be insecure, inefficient, unscalable, and poorly designed. As municipalities begin to include Internet of Things (IoT) and Cyber-Physical Systems (CPS) as part of their Smart-City offerings, it becomes all the more important to ensure that deployments are secure and well regulated. This concern is demonstrated by Rolf Weber's remarks on the legal requirements of IoT in his 2010 paper: "The nature of the IoT asks for a heterogeneous and differentiated legal framework that adequately takes into account the globality, verticality, ubiquity, and technicity of the IoT." [1] Efforts should be taken by technologists to enable legislators to make decisions that are flexible and comprehensive in regards to ICT. Efforts from standardization

organizations are underway to discover and disseminate best practices among Smart Cities. In particular, the National Institute of Standards and Technology (NIST) "International Technical Working Group on IoT-Enabled Smart City Framework" [2] is a multi-faceted approach to establishing and distilling the common features of existing Smart-City architecture and implementations. These efforts, though, require a physical manifestation to validate best practices in real-world scenarios. Siloed or isolated implementations suffice as demonstrations of the effectiveness that IoT-driven Smart City technologies can achieve. However, a collaborative and replicable architecture, which can be deployed on-demand, can ease the transition for technology adoption and maintain standards-based technology decisions. One of the primary goals of NIST Global City Teams Challenge (GCTC) is to enable collaboration among different cities on smart city solutions that are replicable, scalable, and sustainable. Such solutions can accelerate adoption and increase the economy of scale. NIST established the Replicable Smart City Technologies cooperative-agreement program to provide funding to enable cities to participate in team-based GCTC efforts to pursue and implement replicable standards-based smart city solutions. To that end, this paper introduces a collaborative environment for creating a replicable, flexible, secure, and standards- based IoT architecture for Smart Cities. This project seeks to meet the following goals

(1) Delineate Guiding Principles: Design a reference architecture based on principles established through interviews of Smart City stakeholders, technology practitioners, academic researchers, and policy makers.

(2) Introduce Foundational Assets of the Framework: Based on the guiding principles, this work describes the components of an architecture which enables flexible and secure multi-faceted ICT at scale.

(3) Demonstrate the Existing Functionality: Existing architectures are explored in a base-level implementation on a popular cloud platform.

(4) Call for Participation: The most important component of the work is the dissemination of an example architecture to other community and municipal partners for collaboration. Participants within the GCTC have generally agreed to replicate an example architecture and contribute to the environment. However, collaboration with the larger Smart City community would enable a more modular approach that is flexible enough to accommodate more unique use cases.

2 RELATED WORK

The idea for a network of connected devices using machine-to-machine communication has been a point of emphasis since the early 1990s. In discussing his view of the "Computer of the 21st century" [3], Mark Weiser envisioned "pads and tabs" that would change the way interaction with the cyber domain would occur. The "pads" from his view are similar in many respects to consumer Smart devices such as tablets and Smartphones; while "tabs" are the embedded devices (e.g. sensors and actuators) that can be added to physical devices to enable cyber interaction. In this view, IoT is the union of these interconnected networks; a vast web of embedded

and user devices communicating with one another in real-time to enable cyber-physical interactions [4].

In practice, these networks are often standalone, highly siloed, and vertically integrated, lending credence to the term "Intranet of Things" [5]. While this modality is acceptable or even preferred at the residential level, the Smart City will invariably require a higher level of interoperability and integration of systems [6]. Attempts to describe a sufficient architecture of the Internet of Things have taken varied approaches, including representing systems from high-level abstractions including the human nervous system [7] or social organization[8]. Standards organizations such as EPCGlobal have focused on unique identification [9]. Researchers have set out to describe relationships among devices with precise ontologies (e.g. IoTLite Ontology [10], Ontologies for the Internet of Things [11], The semantic smart gateway framework [12]). Unique addressing and the specificity provided through ontologies help to bridge the communication between devices and networks.

Recent efforts in the Smart City community have led to greater cooperation between municipalities. The SmartAmerica Challenge [13] and the GCTC [14] encouraged collaboration among constituent members to produce new cyber-physical systems approaches to smart governance. The ICT deployment of these developments have, to this point, been disjoint. This work, in the spirit of the above challenges, seeks to (1) increase the ability for Smart Cities to collaborate through the establishment of a replicable Smart City architecture, and (2) provide a call for participation in the extension and creation of new assets through an online repository.

3 GUIDING PRINCIPLES

Currently existing Smart City architectures vary wildly in nearly every parameter that can be imagined (e.g. scale, data velocity, security, etc...). Part of this is by design, with exploratory deployments that would induce a lot of overhead and potential security concerns if merged with existing municipal ICT. However, many differences are due to a lack of convergence and standardization in the application of CPS. As these deployments mature and rates of CPS adoption increase within Smart City offerings, there will be challenges which must be overcome.

An architecture that has emerged from the GCTC efforts contains four basic layers–hardware (e.g., sensors and actuators), communications (e.g., wired and wireless), data analytics (e.g., data storage and processing), and service (e.g., decision making) where hardware and communications are part of the infrastructure and service and data analytics are part of the applications [14]. This particular approach (1) provides a methodology for analyzing implementations based on what is being deployed at the different layers and (2) enables a means to identify best practices from current deployments. In this work, we sought to select heuristics that a Smart City system should seek to use in the creation of a secure, flexible, and replicable architecture. We propose the following rules as our "Guiding Principles."

  (1) Vendor and technology neutral - to enable solutions that have broad application, avoid vendor lock-in, and allow new vendors and technologies to be integrated over time.
  (2) Implementable in cloud, on-premise or hybrid computing environments - to facilitate replication across a variety of

infrastructure choices as well as enhance scalability and portability of data.

(3) Leverage open source software when possible - to take advantage of software assets, Application Programming Interfaces, and best practices from the development community.

(4) Small set of core standards/protocols augmented with application-specific standards/protocols - to provide a reusable foundation of proven technologies and protocols that can be augmented when needed to accommodate special cases.

(5) Components and services / microservices - to enable modularity, reusability, scalability and testability in the design and implementation.

(6) Flexible and extendable data interchange formats - to provide standard ways of representing and exchanging data and allow application-specific extensions when needed.

(7) Facilitate data on-ramps / off-ramps - to facilitate the integration of additional data producers and data consumers as needs and technology solutions evolve and scale.

(8) Build in security and privacy - to address security and privacy considerations from the beginning rather than as an afterthought.

4 FOUNDATION

The creation of a replicable Smart Cities architecture is an ongoing project. The assets that have been created thus far are based on the guiding principles delineated in section 3. Further, to foster standardization and increase replicability of the implementation, we have identified several architectural patterns that represent the building blocks of our architecture. These patterns are discussed broadly in 4.1 and more specifically in 4.2.

4.1 Basic Interaction Styles

There are a number of different ways systems and applications can interact, with a multitude of fine-grained technical attributes that describe detailed behaviors and properties. At the highest level, three basic interaction styles have been chosen to describe architectural patterns:

- Real-Time
- Store and Forward
- Batch

The choice of a small number of easily understood interaction styles facilitates communication among the business and technical members of the team as well as across teams having different levels of technical depth.

Real Time: A real-time interaction is one where all parties have a general expectation that something will happen *"right now." Right now* is a loose term, but in general it means that something will happen on the order of seconds or less rather than days or weeks. Real-time interactions are often (but not always) short exchanges involving events or data. The letter R is used to signify a real-time interaction.

Store and Forward: A store-and-forward interaction is one where the sending and receiving sides of an interaction may operate at different rate; and, a technical mechanism in-between them smooths out the interaction. The general expectation is that things sent

by the sender will safely accumulate until the receiver is able to process them. The time delay between sender and receiver could be milliseconds, days, weeks or even years. Email messages are an example of store-and-forward behavior. The letter F is used to signify a store-and-forward interaction.

Batch: A batch interaction is one where groups of information elements are exchanged at the same time in batches. A batch could be as small as one item or as large as billions of items. Nevertheless the general idea is that some number of elements are collected together and transmitted as a group without the expectations of a real-time interaction. Batch interactions are often (but not always) larger exchanges involving collections of events or data. The letter B is used to signify a batch interaction.

4.2 Foundational Interaction Patterns

Patterns are repeatable ways of organizing hardware, software and systems. They make it easier to recognize when a particular design goal can be implemented using a design that has been implemented before. There are many options, variations and details associated with any given technical architecture and this work is not attempting to describe every permutation. Instead, the aim is to describe the primary patterns that can be used to quickly differentiate, characterize and communicate how a particular replicable implementation works at the conceptual level. The five patterns described below are depicted in Figure 1.

(a) Send data from an edge device to one or more data consumers through a message broker

Data at the edge will (almost invariably) be distributed to one or more consuming applications. The data may be raw data, compressed data, or even encoded events depending on the computational power of the device and the communications capabilities of the sensor. The data often needs to be distributed to multiple consuming endpoints. The use of a message broker eases this process. The broker itself can be maintained in the cloud or on-premises through a smart gateway to enable reduced latency on machine-to-machine communications at the edge.

(b) Send message to edge device through a message broker

Actuators at the edge often receive control signals from remote applications and sensors may have special commands for calibration or other purposes. The use of a message broker enables the authentication and management of commands to be sent to the device in a secure and standards-based fashion.

(c) Send data from edge device through existing API

Many existing devices and their supporting applications are developed with an existing Application Program Interface (API). These APIs may be open or proprietary to a particular vendor and consuming applications interact with the API to transfer data and events.

(d) Send message to edge-device through existing API

Similar to the production of data through an API, some edge devices, which have a control or configuration interface, must be accessed through their native API. A controlling application interfaces to the native API to communicate with the edge device.

(e) Bridge data across disparate cloud instances, applications or vendor systems

Many Smart City environments are composed of more than one

(a) Produce/consume data through MQTT

(b) Send Message through MQTT

(c) Produce/consume data through API

(d) Send message through API

(e) Two- and Three-way bridging between disparate cloud environments
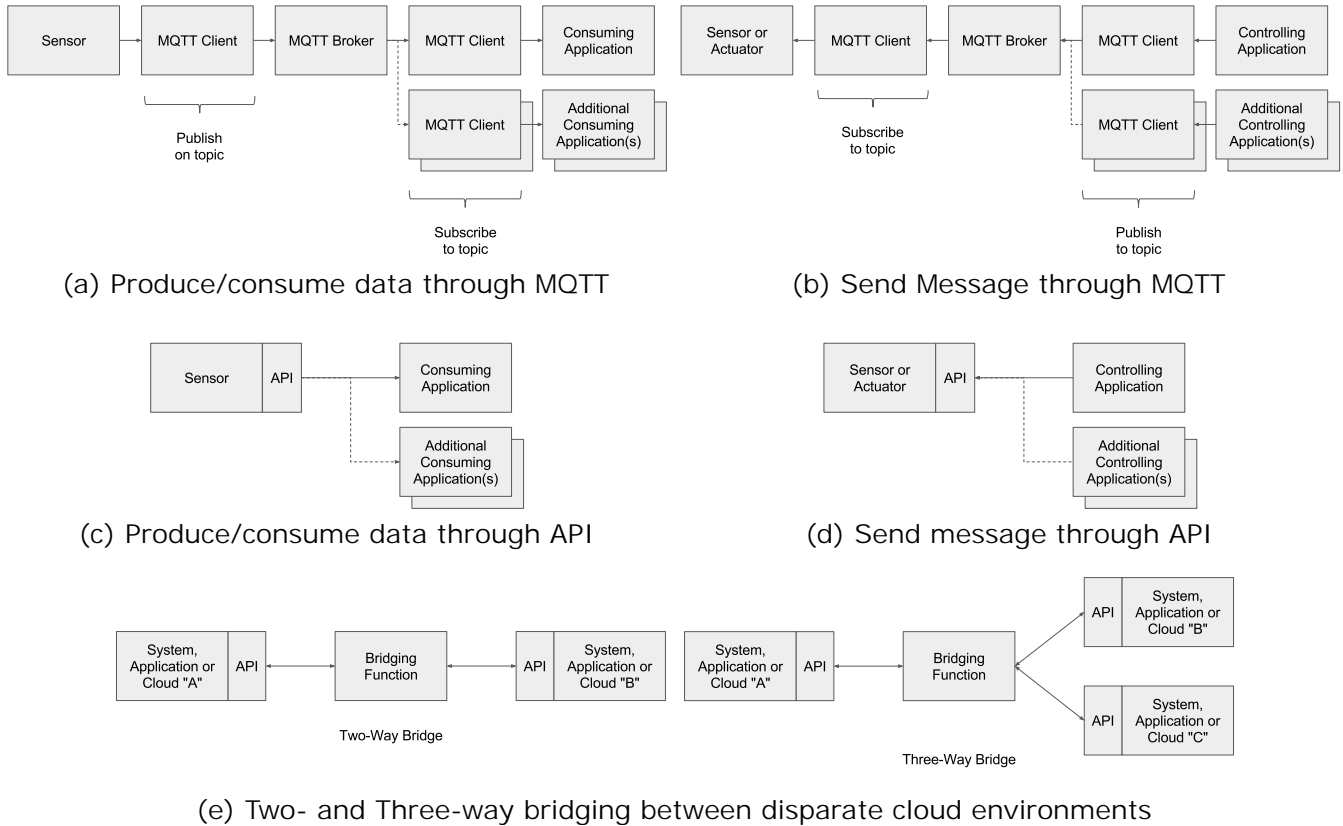
Figure 1: This figure demonstrates architectural patterns that are common, which represent the foundation of signal dissemination within our ecosystem. Sensors and actuators which can communicate over a publish/subscribe message broker schema such as MQTT follow the patterns in (a) and (b) for inbound and outbound traffic. Devices which are prebuilt with their own API (patterns (c) and (d)) should specify their data production and message consumption interfaces, which are controlled by applications either in the cloud or on-site. For environments that involve multiple cloud providers or vendor systems, bridging functions such as the two-way and three-way bridges shown in (e) are developed to tie the systems together.

system or application. Heterogeneous environments may contain cloud-based applications, on-premise applications, mobile and web applications, and vendor applications that run on the vendor's infrastructure. Bridging functions are used to tie these systems together. A bridging function is an application program or service that knows how to interface to the APIs of each system being integrated. It may perform authentication between systems, control authorization of capabilities, translate data from one format to another, consolidate or distribute data and events, provide synchronization and buffering, or perform other functions that allow different producers and consumers to interact and exchange information and commands. Interaction styles may include real-time, store-and-forward, or batch depending on the specific systems being tied together. Bridging functions perform a key role in linking together disparate systems in heterogeneous Smart City environments.

5 REAL WORLD EXAMPLE

To demonstrate the effectiveness of the above design choices, a set of three use cases have been developed at the Montgomery County Thingstitute. These use cases comprise disparate data types and

data usage patterns in geospatially separated environments. The three use cases are depicted in Figure 2. The hardware utilized for each of the Use cases is depicted in Figure 3.

Smart Transportation GPS Tracking: The first use case utilizes a cellular enabled GPS tracking device deployed in municipal vehicles. The system leverages the ISO 6709 GPS coordinate system as its datatype. The usage pattern is for real-time dissemination to administration staff and/or riders of a public transportation system.

Our implementation uses a MachFu MACHGateway, a smart gateway that combines a long-term evolution (LTE) radio with an application interface that is programmable through Android Studio. The combination of the application interface and the wireless personal area network (WPAN) radio access allows for eventual extension of this gateway into an edge processing unit that follows a Fog Computing pattern [15]. The device communicates to our cloud instance in Microsoft Azure through the MQ Telemetry Transport (MQTT) application layer. Thus, this use case follows interaction pattern (a) as defined in Section IV-B. The transmitted data is to be used in a real-time tracking system for the Montgomery County
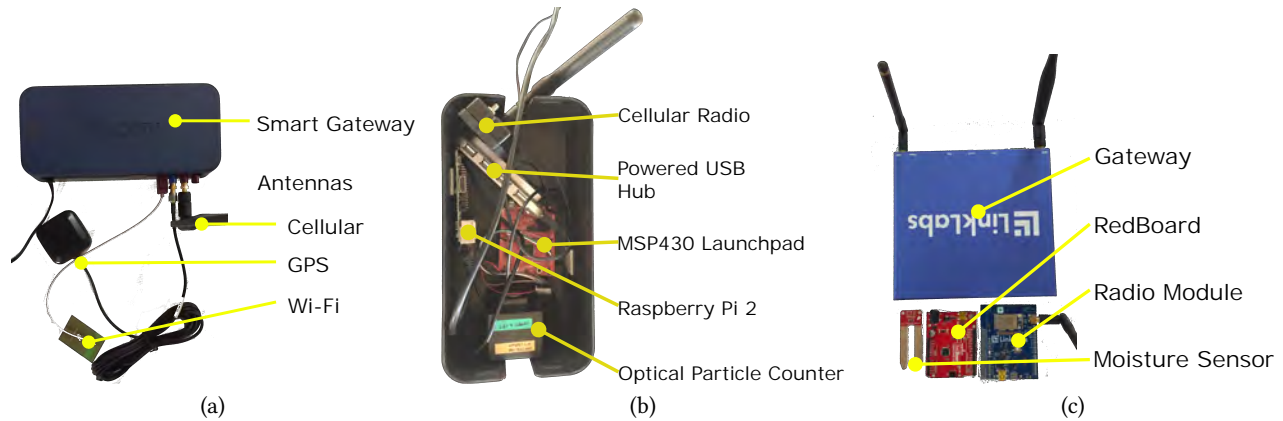
Figure 3: Example hardware used to demonstrate interaction patterns. (a) Smart Transportation hardware using MachGateway as the GPS collector and LTE radio. (b) Optical Particle Counter and supporting hardware to use mixed-mode WAN interfaces. (c) Soil moisture sensor and hardware to sample and send data through a LoRa radio network.
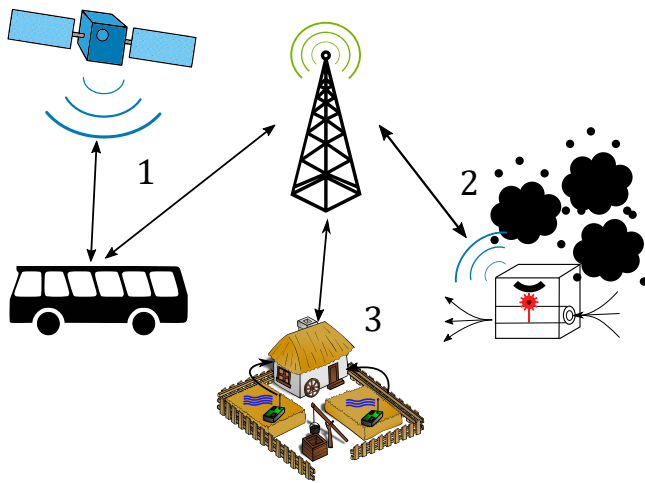

Figure 2: Representation of the three use cases. (1) GPS tracking of transportation assets. (2) Optical particle counter. (3) Smart-Agriculture soil-moisture micro-grid. These three systems represent disparate environments, data types, and connectivity options which demonstrate the need for a flexible Smart-City architecture.

Department of Transportation and follows the real-time interaction style.

Multi-resident Building Optical Particle Counting: The second use case aims at Health and Human Services needs through the use of optical particle counters in multi-resident buildings such as apartment complexes or senior-living facilities. The devices are deployed to sense and report on aberrant particle counts, which can result in harmful health environments, especially for those with existing health constraints such as Chronic Obstructive Pulmonary Disease (COPD) and asthma.

This use case is implemented through a Raspberry Pi acting as the gateway, which communicates through Ethernet, Wi-Fi, or cellular. The OPC itself is an Alphasense OPC-N2, which exposes an SPI interface. An MSP430 Launchpad (F5529) is used to collect data from the OPC and relay that to the Raspberry Pi through an emulated serial interface. The Raspberry Pi then caches the data, and forwards it through one of the WAN interfaces to the cloud. In this use case, the data interaction style is Store and Forward, since the information is not immediately needed for decision making. The communication pattern to the cloud follows Pattern (a) by sending data through a message broker. However, at the edge, the pattern can be seen as following Pattern (b) as the OPC has an SPI API, and the MSP430 wraps that API and acts as a serial forwarder to the Raspberry Pi. The data itself follows the Batch interaction style.

Smart Agriculture Micro-Grid: The last use case seeks to enable micro-grid sensing of agricultural data for the connected farm. Higher crop yield and land utilization within mixed rural and urban municipalities is a driver in localized food communities and can support locally grown farm-to-table organic markets and restaurants.

Our implementation considers that the farm environment is unlikely to maintain WLAN connectivity far from the main facility. We therefore use a LoRa connectivity solution through a Link Labs gateway and radio set. Link Labs implements a LP-WAN interface called Symphony Link that allows messages and data to be passed through these low-power devices to a local gateway, which then forwards the data to a central server through a WAN interface. For the soil-moisture sensor, we use a capacitive sensor that sticks into the soil and measures the permittivity of the field to approximate the moisture content. The capacitor sensor is sampled by a SparkFun RedBoard, and a Link Labs Arduino shield is used to forward the data to the Link Labs cloud where it persists for some amount of time. The data can then be moved to a separate cloud instance through a bridge. This implementation then follows architectural Pattern (e) by using a bridge to communicate with the end sensor. The data follows the Store-and-Forward interaction style.

6 CONCLUSION

This work discusses an implementation of a replicable IoT enabled Smart City architecture. This initial investigation is sufficient to demonstrate the flexibility of the patterns and components that we have identified in consuming and disseminating information that is representative of potential Smart City developments. However, the success of this work ultimately depends on (1) how well the patterns and architectural constructs can scale and (2) whether it can be replicated to other municipalities. Participants within the GCTC have agreed to replicate the patterns and constructs, participate in their development, and contribute to the overall environment including potential additional examples. To aid in this, an online hub and repository has been established in which to exchange ideas, code, and generally collaborate in creating a flexible, scalable, and, most of all, replicable IoT enables Smart City architecture.

The site and the resources associated with this project can be accessed at *https://replicablesmartcities.github.io/*. The site gives information on the project, the ideas set forth in this document, and links to code repositories and other resources which will be added as the project moves forward. Those who wish to participate in the development process are invited to become members of the repository.

ACKNOWLEDGMENTS

DISCLAIMER

Certain commercial products are identified in order to adequately specify the procedure; this does not imply endorsement or recommendation by NIST, nor does it imply that such products are necessarily the best available for the purpose.

Official contribution of the National Institute of Standards and Technology; not subject to copyright in the United States.

REFERENCES

[1] R. H. Weber, "Internet of things–new security and privacy challenges," *Computer Law & Security Review*, vol. 26, no. 1, pp. 23–30, 2010.

[2] National Institutes of Standards and Technology, *IoT-Enabled Smart City Framework*, 2016 (accessed February 8, 2017). [Online]. Available: https://pages.nist.gov/smartcitiesarchitecture/

[3] M. Weiser, "The computer for the 21st century," *Scientific american*, vol. 265, no. 3, pp. 94–104, 1991.

[4] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[5] M. Zorzi, A. Gluhak, S. Lange, and A. Bassi, "From today's intranet of things to a future internet of things: a wireless-and mobility-related view," *IEEE Wireless Communications*, vol. 17, no. 6, 2010.

[6] T. Nam and T. A. Pardo, "Conceptualizing smart city with dimensions of technology, people, and institutions," in *Proceedings of the 12th annual international digital government research conference: digital government innovation in challenging times*. ACM, 2011, pp. 282–291.

[7] A. Liotta, "The cognitive net is coming," *IEEE Spectrum*, vol. 50, no. 8, pp. 26–31, 2013.

[8] H. Ning and Z. Wang, "Future internet of things architecture: like mankind neural system or social organization framework?" *IEEE Communications Letters*, vol. 15, no. 4, pp. 461–463, 2011.

[9] K. Traub, G. Allgair, H. Barthel, L. Burstein, J. Garrett, B. Hogan, B. Rodrigues, S. Sarma, J. Schmidt, C. Schramek *et al.*, "The epcglobal architecture framework," *EPCglobal Ratified specification*, 2005.

[10] M. Compton, P. Barnaghi, L. Bermudez, R. GarcíA-Castro, O. Corcho, S. Cox, J. Graybeal, M. Hauswirth, C. Henson, A. Herzog *et al.*, "The ssn ontology of the w3c semantic sensor network incubator group," *Web semantics: science, services and agents on the World Wide Web*, vol. 17, pp. 25–32, 2012.

[11] S. Hachem, T. Teixeira, and V. Issarny, "Ontologies for the internet of things," in *Proceedings of the 8th Middleware Doctoral Symposium*. ACM, 2011, p. 3.

[12] K. Kotis and A. Katasonov, "Semantic interoperability on the web of things: The semantic smart gateway framework," in *Complex, Intelligent and Software Intensive Systems (CISIS), 2012 Sixth International Conference on*. IEEE, 2012, pp. 630–635.

[13] J. Zander and P. J. Mosterman, "From internet of things through computation of things to a prediction engine-smartamerica challenge 2013 workshop at white house," *Washington DC, Dec*, 2013.

[14] S. Rhee, "Catalyzing the internet of things and smart cities: Global city teams challenge," in *Science of Smart City Operations and Platforms Engineering (SCOPE) in partnership with Global City Teams Challenge (GCTC)(SCOPE-GCTC), 2016 1st International Workshop on*. IEEE, 2016, pp. 1–4.

[15] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*. ACM, 2012, pp. 13–16.

[16] K. Benson, C. Fracchia, G. Wang, Q. Zhu, S. Almomen, J. Cohn, L. Dfiarcy, D. Hoffman, M. Makai, J. Stamatakis *et al.*, "Scale: Safe community awareness and alerting leveraging the internet of things," *IEEE Communications Magazine*, vol. 53, no. 12, pp. 27–34, 2015.