Design Patterns for the Industrial Internet of Things

Gedare Bloom*, Bassma Alsulami*†, Ebelechukwu Nwafor* and Ivan Cibrario Bertolotti‡
*Electrical Engineering and Computer Science, Howard University, Washington, DC, USA
gedare.bloom@howard.edu, bassma.alsulami@bison.howard.edu, ebelechukwu.nwafor@bison.howard.edu
†Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia
balsulami@kau.edu.sa
‡IEIIT, Italian National Research Council (CNR), Torino, Italy
ivan.cibrario@ieiit.cnr.it

***Abstract*—The Internet of Things (IoT) is a vast collection of interconnected sensors, devices, and services that share data and information over the Internet with the objective of leveraging multiple information sources to optimize related systems. The technologies associated with the IoT have significantly improved the quality of many existing applications by reducing costs, improving functionality, increasing access to resources, and enhancing automation. The adoption of IoT by industries has led to the next industrial revolution: Industry 4.0. The rise of the Industrial IoT (IIoT) promises to enhance factory management, process optimization, worker safety, and more. However, the rollout of the IIoT is not without significant issues, and many of these act as major barriers that prevent fully achieving the vision of Industry 4.0. One major area of concern is the security and privacy of the massive datasets that are captured and stored, which may leak information about intellectual property, trade secrets, and other competitive knowledge. As a way forward toward solving security and privacy concerns, we aim in this paper to identify common input-output (I/O) design patterns that exist in applications of the IIoT. These design patterns enable constructing an abstract model representation of data flow semantics used by such applications, and therefore better understand how to secure the information related to IIoT operations. In this paper, we describe communication protocols and identify common I/O design patterns for IIoT applications with an emphasis on data flow in edge devices, which, in the industrial control system (ICS) setting, are most often involved in process control or monitoring.**

***Index Terms*—Industrial Internet of Things, IIoT, Design Patterns, Industry 4.0**

I. Introduction

Kevin Ashton (1999) first coined the phrase IoT for a system of linked devices. IoT has taken the initial advances of Internet technology to a new level, whereby every object in our environment will ultimately have its own unique identifier and be connected to all the other objects around us. Everyday goods, such as TV sets, fridge freezers, automobiles, and even clothing, will amass data regarding the ways in which they are being used; this data will be passed around the IoT, and devices will be instructed to behave in the most efficient and user-friendly fashion as determined by the analysis of that data [1]. Progress on the IoT has been rapid and is growing exponentially. By the year 2020, Gartner predicts that there will be 25 billion unique devices attached to the global IT infrastructure [2].

The fundamental basis of IoT is that many different devices have been set up so that they can be interrogated and manipulated via the Internet by human users themselves or by programs that mirror the aims and desires of those users. The IoT is already having a transformative effect on the way human beings interact, not only with their environment but also with each other. The ways in which we work, with our houses, vehicles, civil services, shops, factories, even weapons, will be changed dramatically. Healthcare, education, and resources will be offered in a swifter, more efficient fashion that is personalized to the consumer [1]. Companies like Walmart are already using radio-frequency identification (RFID) tags to manage their stock; this is an example of very basic IoT implementation [3].

As the number and complexity of connected devices grows, so does the chance of security vulnerabilities that can be exploited by hackers who attempt to manipulate connected devices to their own advantage. As with traditional computing systems, the majority of IoT security attacks will be launched through software; the fact that many different appliances will operate using very similar software makes it ripe for malicious actions to propagate and become widespread [4]. As such, making the IoT secure will be of fundamental concern.

Along the years, a flurry of research work has been aimed at applying IoT concepts in industrial control system (ICS) environments [5], to the point that industrial IoT (IIoT) is nowadays considered one of the pillars of Industry 4.0 [6]. Bringing IoT concepts into an industrial environment further exacerbates security concerns, because in the industrial setting, security is very often tied to safety, as has also been highlighted in a recent document drafted by the World Economic Forum [7]. For instance, it is conceivable that a security breach

This material is based upon work supported by the National Science Foundation under Grant No. CNS-1646317 and the U.S. Department of Homeland Security under Grant Award Number 2017-ST-062-000003. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Department of Homeland Security.

978-1-5386-1066-4/18/$31.00 ©2018 IEEE

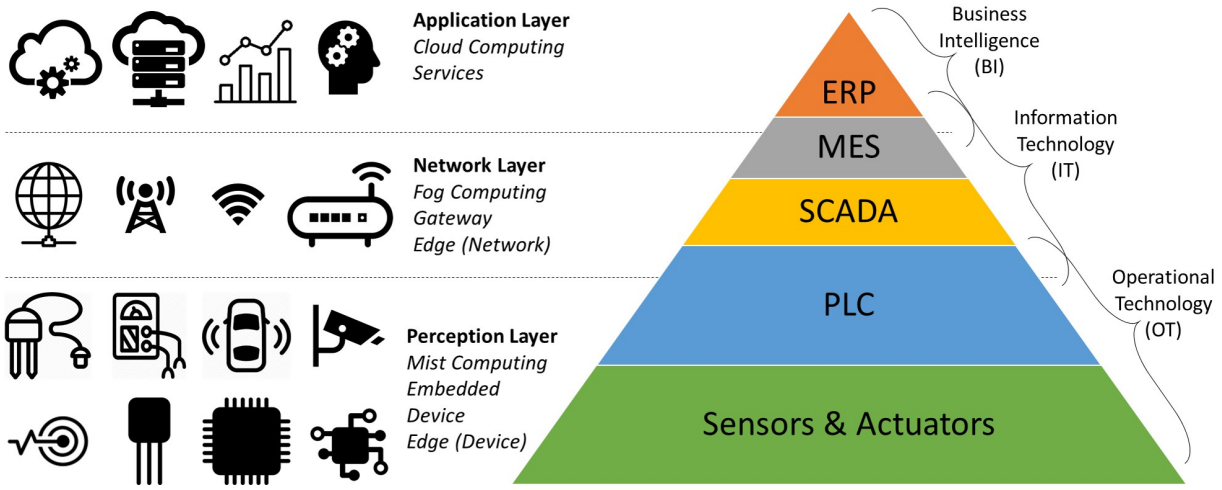


Fig. 1. Hierarchical Layers of the IIoT and Relationship to Pyramid of Automation. Three primary layers are representative of any IoT architecture: perception, network, and application. Industry 4.0 results from the combination of IoT layers and elements with non-integrative manufacturing separated into the pyramid layers starting with the first operational technology layer of sensors and actuators, which are the interface to the physical assets of the shop floor, and moving up through the programmable logic controller (PLC) layer into the information technology layers: the supervisory control and data acquisition (SCADA), manufacturing execution system (MES), and finally the enterprise resource planning in which business decisions are made.

in a factory could easily damage plant machines and lead to physical injury to the human operational or maintenance personnel. Figure 1 shows the mapping of the traditional pyramid of automation with an architectural framework of the IIoT using three primary layers, which are discussed further in Section II. The integration of the pyramid layers by using the IoT layers is the cornerstone of Industry 4.0.

In this paper, we posit that identifying a set of common *design patterns* found in IIoT applications will facilitate security analysis and improve understanding of both realistic threats and feasible security solutions for given design patterns. Design Patterns allow for an abstract model representation that typify design solutions to commonly faced problems. In the IIoT setting, design patterns arise in the commonality of data flow semantics observable across multiple applications. Understanding common I/O design patterns for IIoT applications enables identification of suitable resource allocation, data analytics frameworks, asset selection, and security measures for each pattern. At the same time, according to existing literature [8], staying with proven, well-understood design patterns is very helpful to address and satisfy the real-time constraints and requirements typical of the industrial environment, which are typically more stringent than in traditional IoT. Note that IIoT design patterns differ from those found in the software engineering world, where design patterns focus on the templates of code solutions to recurring problems in software development. Here, design patterns focus on the templates of dataflow solutions to recurring problems in ICS process and business optimization.

The contribution of this paper is the adaptation of design patterns, originally conceived for a different purpose, to the kind of devices and communication protocols typically deployed in the scope of the manufacturing industry with an emphasis on the relevance to factory shop floors. Thus, this paper makes a modest step forward in the path toward fully identifying the IIoT subset of IoT I/O design patterns, which may prove useful to security practitioners and to system designers and engineers alike.

II. Internet of Things (IoT) Background

In this section, we will present an overview of the IoT that describes the layered IoT hierarchical architecture and six requisite elements found in IoT.

A. IoT Hierarchy

IoT is at its essence a layered architecture for distributed systems spanning sensors, microcontrollers, embedded systems, mobile smart devices such as phones and watches, wireless and wired local networking, Internet connectivity, and cloud platforms for durable storage, offloaded computation, and big data analytics. The layers of the IoT each have their own functionality and appliances integral to them. The number of layers used to describe IoT varies [9]–[11], but typically there are at least three main layers that generically are called Perception, Network, and Application. Another common naming scheme for these layers is the embedded, gateway, and cloud, respectively. Recently, the use of mist, fog, cloud is also popular. Edge is occasionally used interchangeably to mean either the edge between the core network of the Internet and the device end-points, or as the edge between the IoT and the "Things" with which the device end-points interact. In addition to the multilayer hierarchy, an effective IoT requires six primary elements: identification, sensing, communication, computation, services, and semantics [12].

1) Perception (Embedded, Edge) Layer: The perception layer, also known as the embedded or the edge layer, is closest to the "Things" of the IoT and is often described in

terms of sensor capabilities. Any sensor input, for example, RFID tags or barcodes, falls into this layer. In many IoT systems, this layer also includes actuators that enable the ability to influence the physical world. The combination of the sensors, actuators, and their associated computational hardware/software is often referred to as an edge device or an IoT node. The responsibilities of the edge device in the perception layer are to collect sensor data from the physical environment, process the data locally—possibly in real-time—and then communicate data with other edge devices or through the network. The communication between nodes in this layer, and occasionally between edge devices and the network layer, varies widely by application domain. Both wireless and wired connections are used to relay information with this layer. Common wireless technologies found at this layer include RF, Bluetooth, ZigBee, 6LoWPAN, and WiFi. Wired technologies include traditional serial connections, such as I^2C and SPI, Ethernet, and application domain-specific technologies such as EtherCAT, controller area network (CAN), digital exchange (DEX), MODBUS, DNP3, and PROFINET.

2) Network (Gateway) Layer: The network or gateway layer facilitates the communication of information provided by the perception layer using wireless, cellular, wired, and Internet network technology. The IoT nodes in this layer are referred to as gateways or hubs. The network layer employs cutting-edge communications technologies to transmit information between the edge and the cloud. Typical technologies at this layer include WiFi, Ethernet, and Cellular. As information is transmitted through this layer, the gateways may filter and aggregate the data in some IoT systems.

3) Application (Cloud) Layer: The application or cloud layer is where the IoT intelligence appears. The practical possibilities of IoT come to the fore at this level by leveraging the vast storage and computational capabilities of cloud datacenters to employ big data analytics on the distributed sensor data produced in the perception layer and aggregated through the network to this layer. The application layer for specific IoT products may consist of one or more public or private clouds. Private clouds have the advantage for their operators of maintaining data control and ownership, which is important for sensitive information such as intellectual property (IP) or personally identifiable information (PII). Their disadvantage is the cost to establish, maintain, and manage them. The upside of public clouds is that cloud vendors are able to provide cheaper computational and storage resources due to effective sharing and economy-of-scale factors in the operation of warehouse computing. The downside is the concern about the trustworthiness both of the cloud provider and, more often, of other clients with whom the cloud resources are shared.

B. IoT Elements

The six elements of the IoT [12], [13] give a better understanding of the nature of applications built for the IIoT.

1) Identification: Correctly matching devices and services requires a means to identify "Things" and link them to digital counterparts. An example of an identification element is an electronic product code. For certain IoT applications to function effectively, it is vital that each device's unique identification and location can be identified. Communication networks need to gather the identity of sensors and their location metadata to incorporate into data collection. For this to work, every device on a network needs a unique identity. Public IP addresses can be, indeed often are, used for identification purposes, but other mechanisms for identification are required when IP addresses are not available or not public. For example, a gateway may have a public IP address, but the edge devices connected to it may be on a private network, and even the sensors could be further separated from the edge devices by other network connections such as wireless or serial wires.

2) Sensing: The sensing element gathers information from objects to transmit for storage or analysis. Once this information has been analyzed, it can be used to generate commands to manipulate how the objects should behave. A myriad of sensors are involved in the IoT, with some examples being wearable monitors for heart rate and pulse oximeter, temperature and humidity, pressure, vibration, chemical pollutants, light, etc. Sensors provide the foundation for intelligence that provides awareness of physical phenomena to the cyber world.

3) Communication: Smart services are offered through the IoT by devices communicating with each other and to cloud platforms. The quality of communication is important, and the IoT becomes less effective when information cannot be transmitted clearly. Suitable communication media change across the layers of the IoT, from low-cost wireless and wired local networks at the perception layer, to high-bandwidth transport networks at the network layer, and the software defined networking (SDN) and network function virtualization (NFV) found in datacenters at the application layer. The heterogeneous and widely varying nature of communication mechanisms across the layers, and even between different nodes within a given layer, is one of the sources of complexity for designing, understanding, and protecting IoT systems.

4) Computation: The computational capacity of the IoT is spread among the hardware and software solutions across the three layers. Traditional microcontrollers are now evolving to offer features specifically for edge devices, for example, Arduino, UDOO, FriendlyARM, Raspberry Pi, Intel Galileo, BeagleBone, Sitara, and WiSense. If the edge devices control physical processes, then real-time operating systems must be employed. At the other end of the IoT stack, datacenters leverage commodity desktop and server hardware typically bundled with a variant of the Linux operating system to provide computational resources that can scale to meet peak demands and pay-as-you-go economic models that are beneficial to new market entrants.

5) Services: There are four main categories of services in the IoT. The primary services are related to identification and are necessary for all other services to work, as they take real-world objects and situate them within a virtual world. The second category of services relates to information aggregation; these services gather, process, and disseminate data. Such

data is then used by the third category of IoT services, the collaborative aware services, which make decisions and issue instructions to edge devices. The final category of services are the ubiquitous services that ensure the three other services are always available.

6) Semantics: In the context of the IoT, semantics means the ability to harvest information from data swiftly and turn-around to provide effective services. Providing effective semantics requires efficiently analyzing data, understanding the relationships among data, and the ability to build models based on non-stationary data sources. Semantic representations are built from technologies such as well-structured data for example XML and JSON, resource description frameworks, and Web ontology languages.

III. Applications of the Industrial IoT (IIoT)

The emergence of an IIoT is revolutionizing nearly all aspects of modern industry. Figure 2 shows four primary categorical areas of IIoT applications: infrastructure, supply chain, process control, and maintenance.

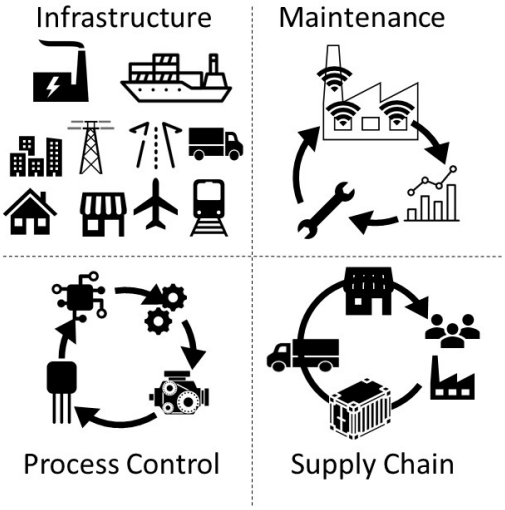


Fig. 2. Categories of IIoT Applications.

A. Infrastructure

Smart devices can make civil infrastructure more flexible, reliable, efficient, and resilient. These improvements can yield great economic advantages for industrial users of the smart infrastructure by providing safety enhancements, reducing costs, and decreasing human effort to maintain and operate infrastructure services. As an example, IoT technology can assimilate data related to energy use and, via smart grids [14], relay that information to the Internet for consumption analysis and advice for reducing costs. Another example is in the growth of smart cities where IoT technology can improve traffic flow, facilitate parking, monitor pollution levels, and more. A smart city can benefit industry by reducing transit costs and simplifying regulatory compliance. Another large market of IoT applications in the infrastructure domain is in the area of smart buildings. Homes and workplaces are now monitored through the technology of IoT. Sensors can monitor energy consumption, operate systems such as lighting, air conditioning and heating, as well as enhance security through surveillance and physical access control [15]. Workplace monitoring and optimization can further reduce the costs associated with ancillary services in support of human labor.

B. Supply Chain

Monitoring sensors such as RFID have been commonplace in supply networks for suppliers, transport companies, and retailers to monitor products as they pass through the supply chain [16]. The IIoT offers the potential to expand the scope of supply chain monitoring techniques far wider by providing opportunities to collect, share, analyze, and track products as they pass through different companies and even across borders between countries. By supplying real-time information concerning the location and transportation of goods, the IoT has the potential to mitigate theft, counterfeiting, and other crimes [17]. The goal of a holistic supply chain management solution is the ability to track products from the production line all the way through decommission; a realistic goal, however, is to track products from production until point-of-sale. For this latter, more practical goal, the integration of RFID from production lines, through the warehouse, in transit, at store-level inventory, and finally at the sales counter will greatly improve business intelligence for product management.

C. Process Control

Another area of great growth in the IIoT is the deployment of sensors in factories to monitor process control and plant state to ensure the factory is operating correctly and to mitigate failures sooner, thus increasing yields and therefore profits. The development of big data analytics to predict future breakdowns is a key enabler for reducing plant downtime. The rise of automation techniques together with monitoring capabilities reduces the cost and risk of human labor in production facilities. A primary challenge to adopt IIoT in plant operations is the need for real-time networks to communicate between sensors, controllers, and gateways. Advancements in real-time wireless sensor networks [18]–[23] and in real-time wired (or cabled) sensor networks [24]–[27] are leading the way forward.

D. Maintenance

Maintenance is a major factor in the efficiency of production lines. Effective maintenance reduces downtime and can decrease energy consumption especially for power-hungry equipment such as motors that can leak massive energy when operated in faulty condition. Traditionally, maintenance approaches fall in two categories: reactive and preventive. Reactive maintenance echoes the sentiment, “if it ain’t broke, don’t fix it.” The run-to-failure scheme of reactive maintenance incurs the least maintenance cost—in the short-term—because of minimal routine downtime and fewer personnel, but has the largest repair costs with both longer downtime due to repairs,

TABLE I
DESIGN PATTERNS AND APPLICATION USE CASES

Design Pattern	Application Use Case
Closed-Loop	Process Control [28], HVAC [29]
Cloud-in-the-Loop	Smart Metering [30], Power Automation [31]
Open-Loop	Wind Turbine Generator [32], Simplex Controller [33]
Cloud-on-the-Loop	Predictive Maintenance [34], Process Optimization [29]
Publisher	Business Optimization [35], Condition Monitoring [36], [37], Asset Tracking [38]
Device-to-Device (D2D)	Distributed Data Cache [39]

unexpected labor costs for those repairs, and, often, more equipment replacement due to permanent failures. Preventive maintenance creates a schedule of maintenance activities that are followed without regard to equipment conditions, which incurs a fixed maintenance and labor cost with fewer emergency repairs and longer equipment life expectancy; on average, preventive maintenance is estimated to save about 12%–18% of total costs in comparison to reactive maintenance [34].

IIoT enables *predictive maintenance* because of the increased knowledge of equipment conditions due to the widespread sensor technology deployed throughout plant floors. Predictive maintenance uses sensors, thermography, data analytics, and human expertise to monitor equipment for vibrations, abnormal temperature and infrared radiation, and lubrication wear in order to schedule maintenance based on predicting problems from early warning signs before they escalate to failures; compared to preventive approaches, predictive maintenance saves on average 8%–12% of total costs, with some estimates as high as 40% savings and a 10x return on investment [34]. The disadvantage of predictive maintenance is that the initial cost to invest in technology needed to gather data for prediction can be high. IIoT helps to reduce some of that cost by leverage commodity sensors and big data analytics to shift the start-up costs.

IV. DESIGN PATTERNS IN THE INDUSTRIAL IOT

No one size fits all architecture easily adapts to the heterogeneous devices and communication protocols used in the IIoT. Therefore, in order to understand complex system designs, a systematic approach of categorizing and controlling heterogeneity can be adopted by system designers, engineers, and operational analysts. Design patterns offer one such approach by identifying reusable components within a system and providing solutions to recurring problems based on those components. As a design aid, theses patterns can be used as a guide by non-domain experts to properly analyze a system by recognizing the patterns.

Our methodology to identify design patterns in the IIoT is guided by the requirements that the patterns must be (a) *abstract* to specific computing devices and communication protocols; (b) *composable* to allow multiple patterns to be easily combined; (c) *recognizable* from typical system components found in the industrial setting; (d) *data-centric* to focus patterns on dataflow semantics as enabling better understanding of resource provisioning and information security requirements. Based on these requirements, we have identified six specific design patterns that appear in the context of applying the IoT to ICS as envisioned by the emerging Industry 4.0:

- Closed-Loop
- Cloud-in-the-Loop
- Open-Loop
- Cloud-on-the-Loop
- Publisher
- Device-to-Device (D2D)

Table I identifies application use cases for each design pattern. In the following, we describe each pattern and highlight one exemplar application of it.

A. Closed-Loop: Classical Closed-Loop Control

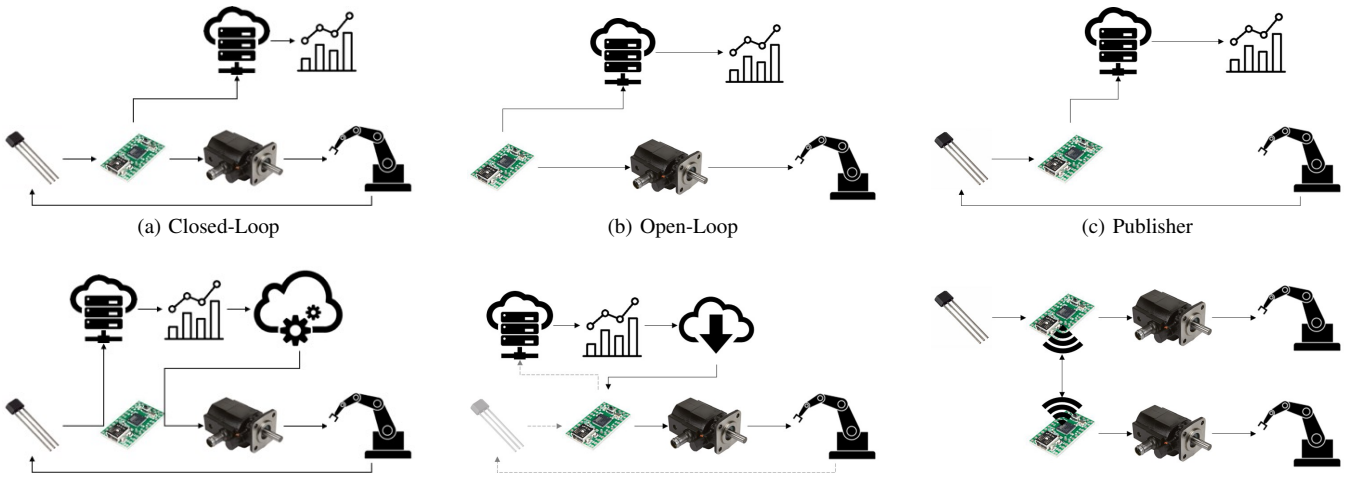
1) Intent: This pattern extracts and stores information from process automation for delayed transmission to the cloud for further analysis without perturbing hard real-time behavior.

2) Motivation: The typical adoption of IoT into industry means bolting on cloud-based data analytics to provide business intelligence. Existing industry automation solutions rely on closed loop control systems to achieve and maintain a predefined output (setpoint) continuously and automatically. The system periodically compares actual condition read from a sensor with desired condition, and generates a control signal based on the difference between the input and the output. According to the control, the actuators will take action.

3) Applicability: Use the Closed-Loop pattern when hard deadlines must be met by the control system that preclude the use of network communications within the path between stimulus (sensor) and response (actuator). This pattern is used to put data collection and communication off the critical path of the control loop.

4) Structure: Figure 3a depicts a prototypical closed loop control system that logs sensor readings to a cloud platform for later analysis.

5) Implementation: The IIoT application for this pattern is remote monitoring of process control. Wireless sensor technology is an ideal terrain for manufacturing controls to keep monitoring measurements of environmental variables such as temperature, humidity, vibration, and many other parameters. According to monitoring of these parameters, actuators will respond based on the sensor measurements. Sensors transmit



(a) Closed-Loop (b) Open-Loop (c) Publisher

(d) Cloud-in-the-Loop (e) Cloud-on-the-Loop (f) Device-to-Device (D2D)

Fig. 3. Design Patterns in the IIoT.

readings to the cloud for analysis, which allows for human operators to monitor the process condition based on the sensing [28].

B. Cloud-in-the-Loop: Closed-Loop Control via the Cloud

1) Intent: This pattern adds a cloud platform as part of a closed-loop control system.

2) Motivation: The addition of cloud computing capabilities—predictive analytics and big data—enable the possibility for cloud platforms, private or public, to be incorporated within classical closed-loop control systems [28].

3) Applicability: Use the Cloud-in-the-Loop pattern when communication with the cloud platform can accommodate real-time analysis to guarantee response times for hard real-time control tasks, or for soft real-time control loops in which occasionally tardy responses from the cloud are permissible.

4) Structure: Figure 3d shows how a closed loop control system could incorporate cloud resources to enhance the computational capabilities of traditional embedded systems.

5) Implementation: A smart meter is an example application with this pattern. While traditional meters record the total consumption, the smart meters are recording the usage and adjust costs according to the time of the day for energy resources. Smart meters include the cloud in the monitoring loop to ensure the flow of information in both directions from the utility company to the consumers, and vice versa [30].

C. Open-Loop: Classical Open-Loop Control

1) Intent: This pattern allows commands to be issued to a process by a local control system unilaterally.

2) Motivation: Although not common in ICS, open loop control is occasionally useful for certain low-cost applications or as a backup control algorithm in case of a fault in the closed-loop system [33].

3) Applicability: Use the Open-Loop pattern to increase dependability of process automation when sensors or external communication links are unreliable.

4) Structure: Figure 3b provides the expected example of open loop control that includes logging to a cloud platform of the actions performed by the controller.

5) Implementation: An example of the Open-Loop pattern is the simpler fail-over open-loop controller used by Simplex control systems [33] that is used when a more complex closed-loop controller is determined to be faulty. In an IIoT setting, logging the actions of the open-loop controller gives insight into the failure rate of primary controllers.

D. Cloud-on-the-Loop: Cloud-configured Control

1) Intent: This pattern migrates supervisory computers to cloud platforms.

2) Motivation: A natural evolution in the IIoT is the emergence of remote management software using cloud platforms to reconfigure control systems. Such systems may be open- or closed-loop control, and the use of the cloud platform enables configuration of the control system parameters. The distinction between cloud-on-the-loop and cloud-in-the-loop is whether the cloud is part of the closed-loop control, i.e., "in" the closed loop, or is observing the open- or closed-loop control and making adjustments as needed, i.e., "on" the loop.

3) Applicability: Use the Cloud-on-the-Loop pattern to analyze data collected by remote monitoring and to send updated configuration data or commands to control systems.

4) Structure: Figure 3e demonstrates the use of a cloud to issue commands to an open-loop controller.

5) Implementation: An Industry 4.0 application that exhibits a Cloud-on-the-Loop pattern is that of automated production optimization, in which factory data is collected and aggregated into a cloud platform that integrates the data into a model and simulation of production to search the parameter space for process optimization, which determines the configuration settings that are relayed back to the process equipment [29].

E. Publisher: Sensor Data Publication

1) Intent: This pattern facilitates data collection from field devices.

2) Motivation: Not all connected devices in IIoT are controllers. Some devices also are responsible for monitoring the plant and reporting sensor measurements by publishing readings to a cloud storage platform.

3) Applicability: Use the Publisher pattern when additional sensing is needed, for example to monitor a plant floor's environment variables or location of humans.

4) Structure: Figure 3c shows how the publisher model fits in the IIoT.

5) Implementation: One of the key applications of the Publisher pattern is in business optimization. As an example, consider energy optimization in production spanning process and facility energy consumption related to shop floor activity [35]. In this application, data collected from process equipment, energy meters on the floor, and facility-wide activities that consume energy, whether proportional to production capacity or not, are aggregated and used to construct parameters for a model to simulate production and predict the impacts of optimization on energy use. Currently, the modeling, simulation, and analysis are carried out in a post hoc manner, thus the data collection uses a Publisher pattern—at multiple points throughout the shop floor—to prepare for optimization.

F. Device-to-Device (D2D): Local Coordination

1) Intent: This pattern enables coordination of decentralized intelligence that leverages shared data among peer machines.

2) Motivation: An emerging application of Industry 4.0 is the ability for machine-to-machine (M2M) communication. Adoption of peer-based M2M enables devices to communicate directly with each, often through short-range, real-time wired or wireless media. The advantage of D2D over traditional networked coordination is that the network layer may be avoided, thus decreasing latency costs for the D2D messages and also congestion for the network backbone.

3) Applicability: Use the Device-to-Device pattern in case systems on the shop floor need to correlate sensor readings as part of a localized yet distributed control system.

4) Structure: Figure 3f demonstrates how devices may communicate with each other.

5) Implementation: An example of D2D for IIoT applications is a distributed data cache for mobile machines that transit the production line and produce (in a Publisher pattern) streaming data that requires low-latency, reliable communications to upload to a cloud platform [39]. As the mobile machines progress through the factory floor collecting data, if their communication link is unreliable then they offload their data to nearby mobile machines or other plant devices using D2D wireless communications, which cache the data and attempt to transmit through the network layer.

G. Pattern Combinations

The designation of patterns does not preclude their overlap—indeed, one of the benefits of naming design patterns is to better understand the nature of a complex system comprising many instances of multiple patterns. Figure 4 illustrates an example combination of patterns in which three control systems are using cloud-in-the-loop, cloud control, D2D, closed-loop, and publisher patterns. An interesting area for future work is in the investigation of tools to facilitate pattern combinations such as an interface description language to compose patterns or a visualization graphical user interface capable of "drag-and-drop" pattern composition.

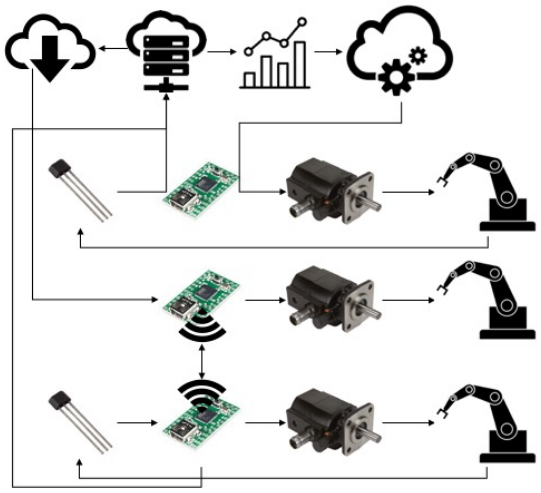


Fig. 4. Combination of Design Patterns: Cloud-in-the Loop, Cloud-on-the-Loop, D2D, Closed-Loop, and Publisher.

V. Correlation with the IIoT Reference Architecture

In the past couple of years, the Industrial Internet Consortium devoted a significant effort to defining the Industrial Internet Reference Architecture (IIRA) [40]. This section briefly presents IIRA's main concepts and outlines how the design patterns proposed in this paper fit within the architectural framework it defines.

A. IIRA Basics

IIRA is an open architecture for IIoT system that, thanks to its fairly high level of abstraction, aims at having broad industrial relevance and applicability while leaving system architects ample design choices. It revolves around the key concept of *viewpoint*, an entity that frames the description and analysis of a specific set of *concerns*. In turn, concerns are aspects or characteristics of a system of interest to *stakeholders*, that is, people or organizations in charge of the system.

Of the four IIRA viewpoints, the *functional* and, to a more limited extent, the *implementation* viewpoint are of interest in this context. The functional viewpoint captures the structure of the functional components of an IIoT system, focusing on the relations, interfaces, and interactions among them, while the

TABLE II
DESIGN PATTERNS AND IIRA FUNCTIONAL DOMAINS

Design Pattern	Pertinent IIRA Functional Domains		
	Control	Operations	Information
Closed-Loop	✓		✓
Cloud-in-the-Loop	✓	✓	✓
Open-Loop	✓		✓
Cloud-on-the-Loop	✓	✓	✓
Publisher			✓
Device-to-Device (D2D)			(✓)

implementation viewpoint is centered around the technology and techniques needed to practically realize them.

With respect to the implementation viewpoint, the main requirement that any IIoT design pattern must satisfy is its compatibility with the architectural patterns that currently drive IIoT system implementations. This is true for the patterns proposed in this paper because they have been drawn according to the generally-accepted IoT architecture definition (Section II), as well as its industry-oriented embodiment presented in recent literature (Section III).

On the other hand, the interrelation between design patterns and the functional viewpoint is more elaborate and deserves a deeper analysis. Table II summarizes which of the IIRA-defined functional domains (within the function viewpoint) are pertinent to each design pattern proposed in Section IV, while the next section illustrates those relationships in more details.

For the sake of completeness, it should also be noted that IIRA currently does not address regulatory and standard-compliance concerns that may arise at the implementation and functional viewpoints. Moreover, IIoT systems may have *crosscutting* concerns, which are crucial because they are related to the safety and security of a system but, at the same time, span multiple viewpoints and require coordination across them. In turn, this may restrict the validity of a viewpoint-by-viewpoint analysis of design patterns.

B. The Functional Viewpoint and Its Functional Domains

Within IIRA, functional domains belong to the functional viewpoint and represent distinct functionalities of an IIoT system. Even though the precise way functional domains are decomposed and decomposition granularity may depend on system-specific requirements, IIRA still identifies and defines five typical functional domains, together with the data flows within and among them. For the sake of brevity, in this paper we focus only on the functional domains especially relevant to the design patterns begin proposed, without further mentioning IIRA's *application* and *business* domains. For the same reason, we do not consider data flows across domains.

1) Control domain: This domain symbolizes all the functions performed by typical industrial control systems (for instance, open- and closed-loop control) and traditionally implemented in proximity of the system they govern. According to IIRA, the control domain is most often characterized by some sort of timing constraints and—in a remarkably close analogy to the inner structure of the design patterns discussed in Section IV—may be further decomposed into sets of functions related to *sensing*, *actuation*, *communication*, and *execution* of the control objectives. Additional sets of functions also belonging to the control domain implement entity abstraction, modeling, and asset management. They are not further discussed here, because they are not embraced by the proposed design patterns.

2) Operations domain: Functions belonging to this domain are mainly in charge of optimizing operations across *multiple* asset types and systems, in contrast with control domain functions, which focus on handling and optimizing one *single* controlled system, like a piece of equipment within a plant. All these functions are inherently related to cloud computing. More specifically, the Cloud-in-the Loop (Section IV-B) and Cloud-on-the-Loop (Section IV-D) patterns can conveniently support three groups of operations domain functions specified by IIRA: *Management* functions, used by asset management centers to issue commands to individual control systems, for instance, to change their setpoint; *Monitoring and diagnostics* functions to detect the occurrence of failures and analyze them; and *Prognostic* functions to identify potential failures before they occur, like is done in preventive maintenance. The ability to compose patterns, outlined in Section IV-G, further streamlines and simplifies the design.

3) Information domain: The information domain is responsible for collecting data from other domains and operating on them to gain higher-level information about overall system performance and behavior, with the help of predictive analytics and big data capabilities typical of cloud computing. Since all design patterns proposed in this paper, with the exception of Device-to-Device, directly contribute to this kind of data collection, they clearly play a role within the information domain. Even the Device-to-Device pattern, when used to cache data and overcome a temporary outage or overload of some communication links (Section IV-F), can be seen as an indirect contributor to data collection.

VI. RELATED WORK

Although considerable work exists on design patterns in software engineering [41]–[43], limited work exists examining design patterns within IoT, and what does exist emphasizes safety-critical systems and a general notion of IoT. To the best of our knowledge, this paper is the first attempt at categorizing design patterns for IIoT systems.

Ashraf [44] proposed fourteen design patterns for safety-critical embedded systems with functional and non-functional requirements based on safety, reliability, modifiability, cost, and execution time. Koster [45] presents some design patterns for an end to end IoT software architecture through information, interaction, application programming, and use case models. Bruce [46] outlines design patterns based on fault tolerant design methods for real-time embedded systems. Wu et al. [47] presents a template for software design in safety-critical systems. Quanbari et al. [48] outlines four

design patterns (i.e., provisioning, deployment, orchestration and automatic configuration of IoT applications) using tools such as puppet, chef, Docker and Git. Our work differs from all of the aforementioned prior work in that it is focused on networked hardware-software interactions.

VII. Conclusion and Future Work

In this paper, we presented core design patterns for the IIoT in the hopes of fostering more secure and effective device development in an era of complexity and heterogeneity. The technologies associated with the IIoT improve the quality of existing industrial applications by reducing costs, improving functionality, increasing access to resources, and automating production tasks. The emergence of intelligent devices and the IIoT are an integral part of achieving Industry 4.0, and understanding the design patterns behind IIoT development will further reduce the costs associated with adoption of IoT concepts by industry.

While the IIoT undoubtedly offers many advantages, it is not without problems. One of the biggest challenges associated with the increased popularity of the IIoT is that the vast amount of data produced by manufacturing systems, which will be more and more difficult to collect, curate, and analyze. Additional problems are also anticipated in terms of governance, systems management, security, and privacy. We believe that design patterns can provide a framework to use when trying to solve these problems.

Future work will include identifying and sorting additional application use cases into the design pattern categories, developing security mechanisms that are practical for particular design patterns, and creating systems engineering guidance for each pattern. As the IIoT continues to evolve, design techniques may change, and so the design patterns here will need to be extended to accommodate future, unanticipated evolution in the IIoT application space. Moreover, the emergence of fog and edge computing in newer IIoT architectures is going to open new design options related to computational load partitioning, which are worth being investigated and then incorporated into appropriate design patterns as well.

References

[1] R. Mahmoud, T. Yousuf, F. Aloul, and I. Zualkernan, "Internet of things (IoT) security: Current status, challenges and prospective measures," in *Proc. 10th International Conference for Internet Technology and Secured Transactions (ICITST)*, Dec. 2015, pp. 336–341.

[2] Gartner Inc. (2018). [Online]. Available: https://www.gartner.com/technology/home.jsp

[3] J. Wurm, K. Hoang, O. Arias, A. R. Sadeghi, and Y. Jin, "Security analysis on consumer and industrial IoT devices," in *Proc. 21st Asia and South Pacific Design Automation Conference (ASP-DAC)*, Jan. 2016, pp. 519–524.

[4] T. Xu, J. B. Wendt, and M. Potkonjak, "Security of IoT systems: Design challenges and opportunities," in *Proc. IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, Nov. 2014, pp. 417–423.

[5] L. D. Xu, W. He, and S. Li, "Internet of things in industries: A survey," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 2233–2243, Nov. 2014.

[6] M. Wollschlaeger, T. Sauter, and J. Jasperneite, "The future of industrial communication: Automation networks in the era of the internet of things and Industry 4.0," *IEEE Industrial Electronics Magazine*, vol. 11, no. 1, pp. 17–27, Mar. 2017.

[7] World Economic Forum — Center for the Fourth Industrial Revolution. (2017) Industrial internet of things safety and security digital protocol network. Draft. [Online]. Available: http://www3.weforum.org/

[8] S. Carvalho, G. Rossi, and F. Balaguer, "Using design patterns in real time applications," *IFAC Proceedings*, vol. 29, no. 5, pp. 93–96, Nov. 1996, IFAC Workshop on real Time Programming (WRTP).

[9] A. Jain, B. Sharma, and P. Gupta, "Internet of things: Architecture, security goals, and challenges — a survey," *International Journal of Innovative Research in Science and Engineering*, vol. 2, no. 4, pp. 154–163, 2016.

[10] P. Sethi and S. R. Sarangi, "Internet of things: Architectures, protocols, and applications," *Journal of Electrical and Computer Engineering*, vol. 2017, pp. 1–25, 2017.

[11] R. Khan, S. U. Khan, R. Zaheer, and S. Khan, "Future internet: The internet of things architecture, possible applications and key challenges," in *Proc. 10th IEEE International Conference on Frontiers of Information Technology (FIT)*, Dec. 2012, pp. 257–260.

[12] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE Communications Surveys Tutorials*, vol. 17, no. 4, pp. 2347–2376, fourth quarter 2015.

[13] V. Lohan and R. P. Singh, "Research challenges for internet of things: A review," in *Proc. International Conference on Computing and Communication Technologies for Smart Nation (IC3TSN)*, Oct. 2017, pp. 109–117.

[14] M. Yun and B. Yuxin, "Research on the architecture and key technology of internet of things (IoT) applied on smart grid," in *Proc. International Conference on Advances in Energy Engineering (ICAEE)*, Jun. 2010, pp. 69–72.

[15] W. Kuijper and V. Ermolaev, "Sorting out Role Based Access Control," in *Proc. 19th ACM Symposium on Access Control Models and Technologies (SACMAT)*. New York, NY, USA: ACM, 2014, pp. 63–74.

[16] R. Angeles, "RFID technologies: Supply-chain applications and implementation issues," *Information Systems Management*, vol. 22, no. 1, pp. 51–65, Dec. 2005.

[17] D. Simchi-Levi, P. Kaminsky, and E. Simchi-Levi, *Designing and Managing the Supply Chain: Concepts, Strategies, and Case Studies*. New York, NY, USA: McGraw-Hill, 2003.

[18] J. Song, S. Han, A. Mok, D. Chen, M. Lucas, M. Nixon, and W. Pratt, "WirelessHART: Applying wireless technology in real-time industrial process control," in *Proc. IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, Apr. 2008, pp. 377–386.

[19] S. Han, X. Zhu, A. K. Mok, D. Chen, and M. Nixon, "Reliable and real-time communication in industrial wireless mesh networks," in *Proc. 17th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, Apr. 2011, pp. 3–12.

[20] E. Toscano and L. L. Bello, "Multichannel Superframe Scheduling for IEEE 802.15.4 Industrial Wireless Sensor Networks," *IEEE Transactions on Industrial Informatics*, vol. 8, no. 2, pp. 337–350, May 2012.

[21] S. Han, Y.-H. Wei, A. K. Mok, D. Chen, M. Nixon, and E. Rotvold, "Building wireless embedded internet for industrial automation," in *Proc. 39th Annual Conference of the IEEE Industrial Electronics Society (IECON)*, Nov. 2013, pp. 5582–5587.

[22] M. Sha, D. Gunatilaka, C. Wu, and C. Lu, "Empirical study and enhancements of industrial wireless sensor & actuator network protocols," *IEEE Internet of Things Journal*, vol. 4, no. 3, pp. 696–704, Jun. 2017.

[23] K. Yu, M. Gidlund, J. Åkerberg, and M. Björkman, "Performance evaluations and measurements of the REALFLOW routing protocol in wireless industrial networks," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 3, pp. 1410–1420, Jun. 2017.

[24] H. A. Hansson, T. Nolte, C. Norstrom, and S. Punnekkat, "Integrating reliability and timing analysis of CAN-based systems," *IEEE Transactions on Industrial Electronics*, vol. 49, no. 6, pp. 1240–1250, Dec. 2002.

[25] R. I. Davis and N. Navet, "Controller area network (CAN) schedulability analysis for messages with arbitrary deadlines in FIFO and work-conserving queues," in *Proc. 9th IEEE International Workshop on Factory Communication Systems (WFCS)*, May 2012, pp. 33–42.

[26] G. Cena, I. Cibrario Bertolotti, T. Hu, and A. Valenzano, "Design, verification, and performance of a MODBUS-CAN adaptation layer," in

Proc. 10th IEEE Workshop on Factory Communication Systems (WFCS), May 2014, pp. 1–10.

[27] G. Bloom, G. Cena, I. Cibrario Bertolotti, T. Hu, and A. Valenzano, "Optimized event notification in CAN through in-frame replies and Bloom filters," in *Proc. 13th IEEE International Workshop on Factory Communication Systems (WFCS)*, May 2017, pp. 1–10.

[28] T. Samad, "Control systems and the internet of things," *IEEE Control Systems*, vol. 36, no. 1, pp. 13–16, Feb. 2016.

[29] J. Michaloski, F. Proctor, J. Arinez, and J. Berglund, "Toward the ideal of automating production optimization," in *Proc. ASME International Mechanical Engineering Congress and Exposition*, vol. 2A, Nov. 2013, pp. 1–9.

[30] T. Balikhina, A. A. Maqousi, A. AlBanna, and F. Shhadeh, "System architecture for smart home meter," in *Proc. 2nd International Conference on the Applications of Information Technology in Developing Renewable Energy Processes Systems (IT-DREPS)*, Dec. 2017, pp. 1–5.

[31] R. Kumar, M. L. Dewal, and K. Saini, "Utility of SCADA in power generation and distribution system," in *Proc. 3rd IEEE International Conference on Computer Science and Information Technology*, vol. 6, July 2010, pp. 648–652.

[32] H. Polinder, "Overview of and trends in wind turbine generator systems," in *Proc. IEEE Power and Energy Society General Meeting*, July 2011, pp. 1–8.

[33] D. Seto, B. H. Krogh, L. Sha, and A. Chutinan, "Dynamic control system upgrade using the Simplex architecture," *IEEE Control Systems*, vol. 18, no. 4, pp. 72–80, Aug. 1998.

[34] G. P. Sullivan, R. Pugh, A. P. Melendez, and W. D. Hunt, "Operations & Maintenace Best Practices: A Guide to Achieving Operational Efficiency," Aug. 2010. [Online]. Available: https://www.energy.gov/node/907671

[35] J. Arinez, S. Biller, K. Lyons, S. Leong, G. Shao, B. E. Lee, and J. Michaloski, "Benchmarking production system, process energy, and facility energy performance using a systems approach," in *Proc. 10th ACM Performance Metrics for Intelligent Systems Workshop (PerMIS '10)*. New York, NY, USA: ACM, 2010, pp. 88–96.

[36] F. Civerchia, S. Bocchino, C. Salvadori, E. Rossi, L. Maggiani, and M. Petracca, "Industrial Internet of Things monitoring solution for advanced predictive maintenance applications," *Journal of Industrial Information Integration*, vol. 7, pp. 4–12, Sep. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2452414X16300954

[37] M. Saez, F. P. Maturana, K. Barton, and D. M. Tilbury, "Real-time manufacturing machine and system performance monitoring using internet of things," *IEEE Transactions on Automation Science and Engineering*, vol. PP, no. 99, pp. 1–14, 2018.

[38] F. J. Valente and A. C. Neto, "Intelligent steel inventory tracking with IoT / RFID," in *Proc. IEEE International Conference on RFID Technology Application (RFID-TA)*, Sep. 2017, pp. 158–163.

[39] A. Orsino, R. Kovalchukov, A. Samuylov, D. Moltchanov, S. Andreev, Y. Koucheryavy, and M. Valkama, "Caching-aided collaborative D2D operation for predictive data dissemination in industrial IoT," Feb. 2018. [Online]. Available: http://arxiv.org/abs/1802.06902

[40] *The Industrial Internet of Things Volume G1: Reference Architecture*, Industrial Internet Consortium, 2017, IIC:PUB:G1:V1.80:20170131.

[41] P. Coad, "Object-oriented patterns," *Commun. ACM*, vol. 35, no. 9, pp. 152–159, Sep. 1992.

[42] S. Khwaja and M. Alshayeb, "Survey on software design-pattern specification languages," *ACM Comput. Surv.*, vol. 49, no. 1, pp. 21:1–21:35, Jun. 2016.

[43] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns: Elements of Reusable Object-oriented Software*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1995.

[44] A. Armoush, "Design patterns for safety-critical embedded systems," Ph.D. dissertation, RWTH Aachen University, 2010.

[45] M. Koster. (2014) Design Patterns for an Internet of Things. [Online]. Available: https://community.arm.com/iot/b/blog/posts/design-patterns-for-an-internet-of-things

[46] B. P. Douglass, *Real-Time Design Patterns: Robust Scalable Architecture for Real-Time Systems*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2002.

[47] W. Wu and T. Kelly, "Safety tactics for software architecture design," in *Proc. 28th Annual International Computer Software and Applications Conference (COMPSAC)*, vol. 1, Sept 2004, pp. 368–375.

[48] S. Qanbari, S. Pezeshki, R. Raisi, S. Mahdizadeh, R. Rahimzadeh, N. Behinaein, F. Mahmoudi, S. Ayoubzadeh, P. Fazlali, K. Roshani, A. Yaghini, M. Amiri, A. Farivarmoheb, A. Zamani, and S. Dustdar, "IoT design patterns: Computational constructs to design, build and engineer edge applications," in *Proc. 1st IEEE International Conference on Internet-of-Things Design and Implementation (IoTDI)*, Apr. 2016, pp. 277–282.