# Cloud of Things and Interworking IoT Platform: Strategy and Execution Overviews

**Takudzwa Fadziso[1], Harshini Priya Adusumalli[2], Mahesh Babu Pasupuleti[3]**

[1]Institute of Lifelong Learning and Development Studies, Chinhoyi University of Technology, **ZIMBABWE**
[2]Department of Computer Science, Kent State University, Kent, Ohio, **USA**
[3]Data Analyst, Department of IT, Iminds Technology Systems INC, 1145 Bower Hill Rd, Pittsburgh, PA 15243, **USA**

## ABSTRACT

An IoT platform is a multi-layer system that enables linked device automation. IoT platforms are enabling software that connects various hardware devices, access points, and networks to other sections of the value chain. Virtual objects are now essential in any IoT platform. In this work, we design and create a cloud-based IoT platform that allows users to register and initialize virtual objects, then consume them via the IoT marketplace and integrate them into IoT applications. The proposed IoT platform differs from previous IoT platforms in that it provides both hardware and software services on a single platform. The proposed IoT platform is distinct from the IoT marketplace where users can buy and sell virtual things. Based on virtual items in CoT, IoT platform and IoT marketplace experiments are undertaken. The proposed IoT platform is easy to use, secure, and trustworthy. An IoT testbed and a case study for reusing virtual objects in a residential environment are developed. It allows for virtual object discovery and sharing. Virtual objects can monitor and operate IoT devices.

Keywords: Internet of Things (IoT), Cloud of Things (CoT), IoT Marketplace, IoT Platform, Virtualization



## INTRODUCTION

The Internet of Things (IoT) is a network of physical items that collect and share data. IoT is vital in many areas such as transit, education, healthcare, and banking (Truong and Dustdar, 2015). It mixes heterogeneous and versatile items with varying communication and computational capabilities to get insight into a network's physical reality. With IoT, devices can now communicate remotely via smartphones, tablets, or laptops with wired or wireless Internet connections. Smart environments require embedded IoT devices with limited resources (Xia et al., 2012). Network bandwidth and power usage are critical. We introduce low-power and low-bandwidth IoT devices and protocols.

The Internet of Things (IoT) allows developers to connect software to physical devices on a wide scale (Achar, 2017). Smart grids and transportation networks are examples. These applications necessitate real-time controls, big data centers, and fast processing. The internet of things can meet these needs (CoT). CoT provides the platform for IoT device integration (Adusumalli, 2016b).

This is achieved by locating and constructing the most appropriate virtual objects (Dehury and Sahoo, 2016). Context-aware systems must learn system behavior through the constant association of virtual and real-world objects. The web of objects (WoO) virtualizes real-world items using semantic ontology to address these issues.

Previously, scientific application developers (Achar, 2016) had to pick between local workstations, clusters, supercomputers, and networks. Each of these alternatives has various availability, performance, and cost compromises. Cloud computing has recently gained popularity among scientists as a possible alternative for scientific applications.

These days, IoT is confronting various obstacles. The present interaction model is a human–object. Changing the human-object interaction model to object-object interaction is a huge task. Objects will autonomously interact to deliver composite services. Objects must be aware of their context in order to run applications and interact with their surroundings. Integrating the digital and physical worlds is one of computing's key goals. In the age of integrated electronics, we now have smart things. Smart items may communicate with both digital and physical objects.

## LITERATURE REVIEW

In recent years, IoT application implementation has exploded in areas like automotive, utilities, health, logistics, and home automation (Pasupuleti, 2015a). The growing number of devices necessitates developing cloud middleware for handling sensors and actuators. There are open source IoT-Frameworks to distribute and process data (Borgia, 2014). Much study has been done on IoT and cloud computing (Adusumalli, 2017a). The project seeks to bridge a gap highlighted in previous methodologies by combining cloud computing with IoT concepts. Because items are heterogeneous, IoT and cloud computing arose to overcome challenges (Pasupuleti, 2015b). CoT can manage, aggregate, and store data for large-scale IoT platforms (Adusumalli, 2016a).

Researchers are shifting to edge computing. Edge computing meets IoT delay requirements. A distributed computing paradigm where processing is done on dispersed device nodes called smart devices or edge devices rather than a centralized cloud environment (Ray, 2016). FOCAN is a multi-tier architecture in which apps operating on things collaboratively compute, route, and communicate via smart city environment (Adusumalli, 2017b). Kelaidonis et al. provide a framework for RWO virtualization and administration (Achar, 2015). The framework functions on three levels. This enables for clever and adaptable applications at each level. The framework explains the heterogeneity of many devices or objects while boosting dependability and considering multiple users' perspectives. Integrated virtualization has shown to be a cost-effective and time-saving tool (Pasupuleti, 2016b). A stable association between virtual items and real things allows for intelligent service provisioning in IoT infrastructure (Pasupuleti, 2015c). To address these issues, the web-of-objects (WoO) virtualizes real-world items. The learning model is a composite virtual object generated by merging numerous virtual object functions (Adusumalli & Pasupuleti, 2017).

## IOT PLATFORM BASED ON INTERWORKING IOT MARKETPLACE

This portion of the article describes the proposed IoT platform and IoT marketplace architecture. Figure 1 displays the suggested system architecture. The system has three layers: physical, repository, and application service. Sensors and actuators are part of the physical layer. Sensors detect events in the environment, such as temperature, humidity, and pressure. Actuators control a device using a command or signal. Actuators control equipment like lights, fans, and motors. The MQTT protocol connects the physical and repository layers. Physical IoT devices can publish and subscribe to messages. Sensors send data to a MQTT broker, while actuators subscribe to a specific topic.

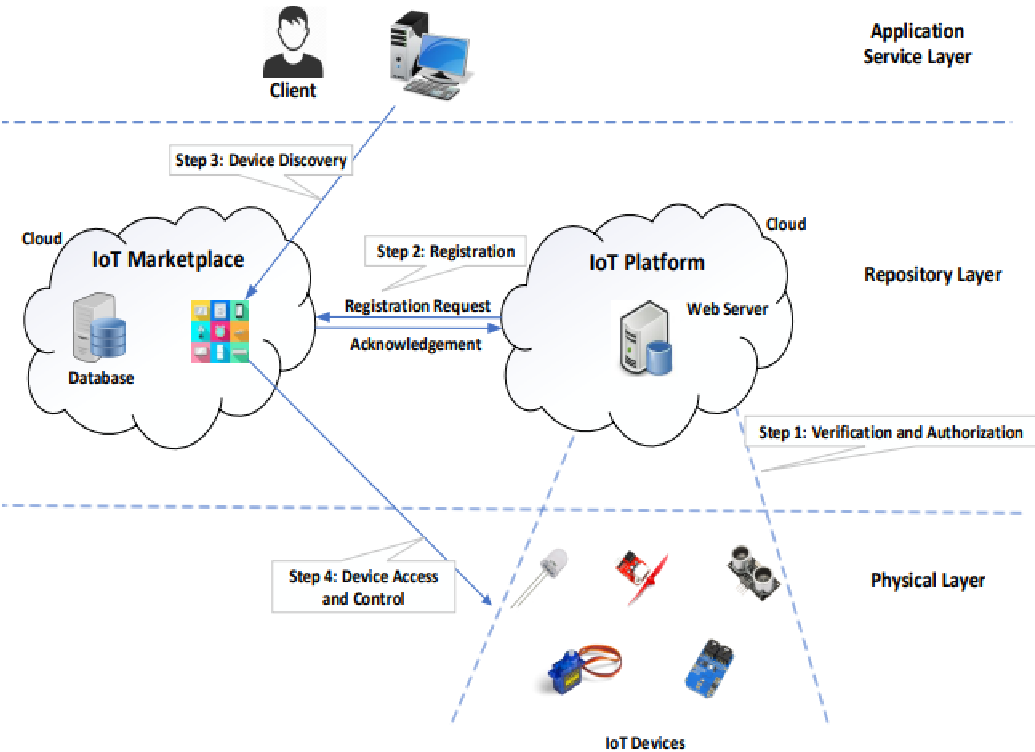


Figure 1: IoT platform and control service mechanism

The repository layer configures, installs, and deploys IoT platforms. Our app was deployed via Amazon Web Services (AWS). The webserver runs on Ubuntu. The IoT device registration data is stored in a MySQL database. Only authorized users can register IoT devices on the market. The registration process verifies an IoT device. The IoT platform receives acknowledgment of registration. The user can adjust the IoT device's visibility to public or private. The application service and the repository layer communicate through HTTP.

IoT Marketplace is accessible via the application service layer's web interface. The application service layer controls the service. The IoT marketplace allows users to quickly monitor and control IoT devices.

## Proposed System Sequence Diagram

The user logs into the client application and inputs their credentials (username and password). Authorization is needed to retrieve data from the server. The server creates an access token after the user is authenticated. The user enters IoT device details including name, type, IP, and MAC. A POST request verifies IoT device meta-data and access token.

The web server responds. If the IoT device is verified, it notifies the client application. After user authorization and IoT device verification, it sends a POST request to the IoT Marketplace through RESTful API. The IoT devices' visibility can now be changed to public or private. IoT Marketplace handles the request and stores the data. It acknowledges the webserver. The client application is notified of successful registration.

Either make the IoT gadget public for anyone to use or private for the owner only. The user first gains access to the IoT marketplace. To log in, the user inputs their user name and password. Invalid credentials mean the user submitted the wrong username or password. Invalid credentials alerts the user. A session is generated for the user who has successfully signed in. IoT devices are listed for the user. They can be monitored and regulated once found. A sensor selected by the user obtains its values, e.g., a BME280 sensor sends temperature, humidity, and pressure values to the IoT marketplace. If the user selects actuator, the actuator must be turned on or off. It will send the command through MQTT to the IoT device. The IoT device will be subscribed to a certain subject. Perform the action and notify the user.

## Configuration of the Proposed System

The proposed system works in five steps: (a) verification and authorization of embedded IoT devices. (b) Registration of the IoT device on the IoT marketplace. (c) Discovery of IoT device. (d) Access to IoT device via virtual object. (e) Data exchange.

### Authorization and Verification

Step 1 authorizes and verifies individuals and IoT devices. Any user can access web services if the URL and input parameter are revealed. To prevent unauthorized access to APIs, the proposed system leverages OAuth 1.0 authentication at checkpoints. OAuth 1.0 requires three steps to authenticate. It starts by sending a POST request to get a request token. The user then submits the token and secret as a query parameter to the IoT platform's authorization URL. If the user agrees, an access token is generated. The access token allows the user to retrieve data from the server. After permission, the user can provide device information such as name, kind, IP, and MAC address and submit a POST request to the web server. The web server verifies the device details.

### Registration of IoT Device

Step 2 involves registering the IoT device on the IoT platform. The registration request is sent to the IoT Marketplace. The IoT devices' visibility can now be changed to public or private. Private mode is for personal use of IoT devices such as monitoring or controlling appliances. Any IoT device can use the public mode. The public can watch but not control IoT devices in public mode (Adusumalli, 2018). Only the device's owner may control it.

### Discovery of IoT Device

Step 3: Find IoT devices. Unregistered users can access public mode IoT devices. In order to find private mode IoT devices, the user must first log in. An IoT marketplace dashboard appears after successful login, displaying registered private and public IoT devices. Users can search for IoT devices. Get the list of registered IoT devices.

### Access to IoT Device

IoT platform control service method (Step 4), it allows access to IoT devices. Only the public can monitor public IoT devices. Except for the owner, public IoT devices are private. The owner of a private IoT device can monitor and operate it from the IoT marketplace dashboard. Sensors and actuators are two categories of IoT devices. The sensor is a device that detects events in its environment and sends the data to other devices for use. Sensors include temperature and motion sensors. An actuator is a machine component that controls a mechanism or system. An actuator needs a control signal and a power source to work. An actuator is a device that allows a control system to act on an environment. Users can watch and read sensor readings, such as temperature from a sensor installed in an environment. For example, users can turn off a fan or turn on an LED. The actuator receives the command via MQTT protocol. The actuator interprets the command and executes it.

**Data Exchange**

RDF is a framework for describing web resources (Pasupuleti, 2016a). RDF is meant to be read by computers. RDF is a web data exchange standard. The fundamental reason for adopting RDF in the proposed system is to facilitate information interchange between machines running different operating systems and application languages. The proposed system can find, share, and store resources. So we used RDF for resource discovery, sharing, and visualization on the IoT marketplace. RDF is a standard for encoding, exchanging, and reusing meta-data. It comprises device ID, kind, and title information. The resource's meta-data can be decrypted and displayed in the IoT market. The IoT marketplace allows servers and clients to effortlessly exchange data using RDF/XML.

## SYSTEM DEVELOPMENT AND TESTING

Figure 2 depicts the suggested system's development model. The IoT platform and IoT marketplace are two sub-systems of the overall system. Both technologies are cloud-based. We used AWS cloud computing. The IoT platform and IoT Marketplace were deployed on the Amazon elastic computing cloud (EC2). The Ubuntu 16.04 OS runs a web server and hosts apps. The suggested system has three layers: application, repository, and context acquisition and control. In the acquisition and control layer, we used two IoT devices (LED and Fan) and one Raspberry Pi as actuators. LED and fan are connected to GPIO pins on Raspberry Pi. The ubuntu 16.04 web server has a MQTT broker installed. A MQTT broker is used to publish and subscribe to messages. Topics might be content-based, topic-based, or type-based. Topic-based is our proposed system. Topic-based has a predetermined set of themes. A broker publishes and subscribes messages. The repository and control layer communicate using MQTT. Because there are so many messages being published and subscribed to, we employed the concept of a distributed broker. The distributed broker relieves the central broker. Various IoT devices are handled by each distributed broker.

The development model employed a Raspberry Pi 3 Model B for context acquisition and control. Actuators include LEDs and fans. The fan is connected to GPIO pin 24. Then they are listed on the IoT Marketplace. Registration requires authorization and verification. Only authorized people need authorization. Authorization requires a username and password. Responding with an access token. For registration, the user submits device details and an access token. Each IoT gadget has a topic. IoT Marketplace sends commands to actuators through MQTT. The command could turn the IoT gadget on or off. The repository layer consists of the configuration, the installation of a web server, and the deployment of the IoT platform and the IoT Marketplace.. The repository layer is responsible for storing the metadata of Internet of Things devices. Communication between the physical layer and the repository layer is accomplished through the MQTT protocol. The registration of IoT devices is accomplished through the usage of RESTful APIs.

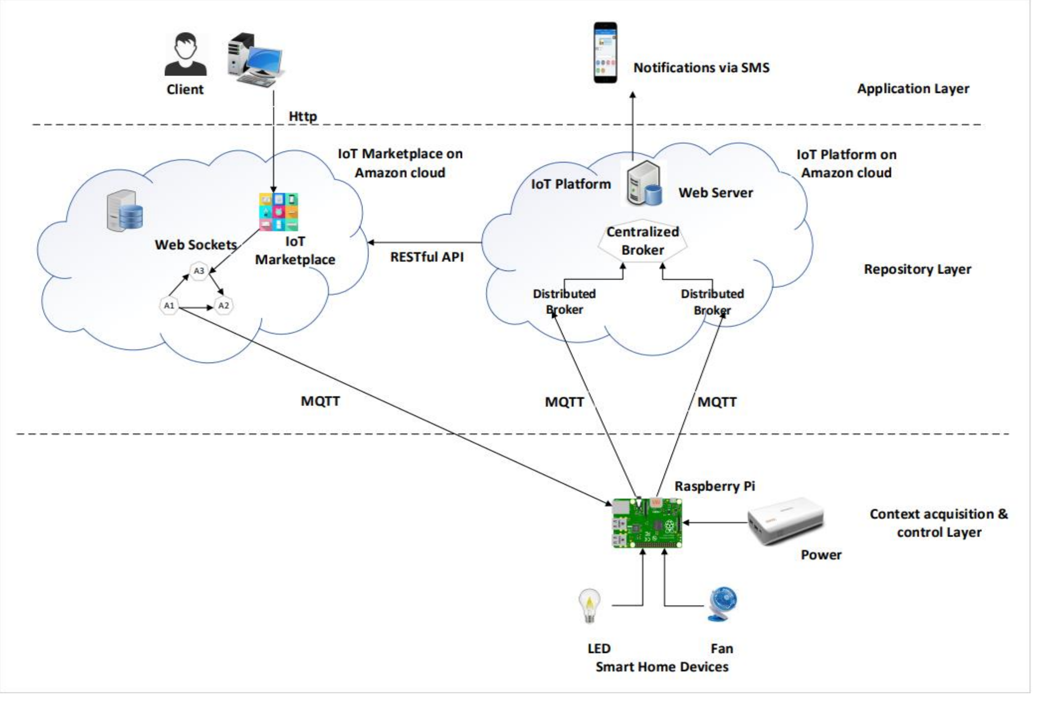


Figure 2: Development model of IoT platform and control service based on the interworking IoT marketplace.

It is possible to access the IoT Marketplace using a user-friendly web interface that is provided by the application service layer. Users can monitor and operate IoT devices through the Internet of Things marketplace. In the following section, you will learn how the Internet of Things platform works in a household environment through the use of a case study.

## EXPERIMENTAL ENVIRONMENT

This section describes the study's experiments. Each IoT device has GPIO pins. GPIO pin 23 is for LED and GPIO pin 24 is for Fan. A cloud web server connects the devices. AWS is a cloud provider. Our solution uses Amazon Elastic Cloud Computing (EC2). The IoT platform and IoT AppStore are installed on Ubuntu 16.04 server. The data is stored in MySQL. The data storage includes registered users, IoT devices, and associated metadata.

WebStorm is a software development tool. WebStorm is used for client-side JavaScript applications. Sublime Editor is used for IoT platform and IoT Marketplace web pages. The Raspberry Pi uses Python to register, connect, and access IoT devices. Thonny is a Python programming tool. FileZila is a program that transfers files from client to server.

Our IoT platform is MVC. This architectural pattern divides an application into three basic logical components: model, view, and controller. These components are designed to handle specific application development tasks. MVC is a popular web development framework for creating scalable and flexible projects. Drupal is an open-source program that anyone can use, regardless of technical ability. It builds and manages dynamic webpages. It can be used to create content management systems. The IoT marketplace is built on Drupal.

## Significance and Judgement

This section will highlight the proposed work's importance and compare it to existing state-of-the-art tools. We compared the proposed IoT platform to existing IoT platforms. With so many types and vendors of IoT platforms, it's hard to know where to start. For example, consumer applications, enterprise IoT applications, etc. The IoT platform market is complex because different IoT projects and software applications have different architectures, modes of connectivity and device management, application development capabilities, and options to leverage IoT.

To compare the performance of IoT platforms, we chose attributes common to all open-source technologies. FIWARE and Kaa are two popular open-source IoT solutions. With these platforms, you can remotely monitor and operate IoT devices. Each IoT platform serves a unique function and provides unique services. The suggested IoT platform can register any sensor or actuator. The IoT gadget can be monitored or operated using the lightweight MQTT protocol. To reduce burden on IoT systems, the suggested method separated the marketplace from the platform (Pasupuleti & Adusumalli, 2018).

## Conclusion

This study proposes an IoT platform and control service based on CoT virtual objects. This article describes how to design and build an architecture that allows the public to register, discover, and access IoT devices. In this study, we created a safe and reliable IoT infrastructure for device registration. For security reasons, registration requires authorization and verification. The authorization process will prevent unauthorized users from registering IoT devices. The model, IP address, and MAC address of an IoT device will be checked to verify its existence. The suggested system also allows for device discovery and sharing. The visibility mode of IoT devices can be configured to public or private. IoT Marketplace helps users find essential IoT devices. The IoT Marketplace allows users to monitor and operate IoT devices. During software tool testing, the proposed system outperforms FIWARE. The proposed IoT platform outperforms FIWARE. Next, we can compare the proposed system's performance to other lightweight communication protocols like CoAP.

## References


Achar, S. (2015). Requirement of Cloud Analytics and Distributed Cloud Computing: An Initial Overview. International Journal of Reciprocal Symmetry and Physical Sciences, 2, 12–18. https://upright.pub/index.php/ijrsps/article/view/70

Achar, S. (2016). Software as a Service (SaaS) as Cloud Computing: Security and Risk vs. Technological Complexity. Engineering International, 4(2), 79-88. https://doi.org/10.18034/ei.v4i2.633

Achar, S. (2017). Asthma Patients' Cloud-Based Health Tracking and Monitoring System in Designed Flashpoint. Malaysian Journal of Medical and Biological Research, 4(2), 159-166. https://doi.org/10.18034/mjmbr.v4i2.648

Adusumalli, H. P. (2016a). Digitization in Production: A Timely Opportunity. *Engineering International*, *4*(2), 73-78. https://doi.org/10.18034/ei.v4i2.595

Adusumalli, H. P. (2016b). How Big Data is Driving Digital Transformation?. *ABC Journal of Advanced Research*, *5*(2), 131-138. https://doi.org/10.18034/abcjar.v5i2.616

Adusumalli, H. P. (2017a). Mobile Application Development through Design-based Investigation. *International Journal of Reciprocal Symmetry and Physical Sciences*, *4*, 14–19. Retrieved from https://upright.pub/index.php/ijrsps/article/view/58

Adusumalli, H. P. (2017b). Software Application Development to Backing the Legitimacy of Digital Annals: Use of the Diplomatic Archives. *ABC Journal of Advanced Research*, *6*(2), 121-126. https://doi.org/10.18034/abcjar.v6i2.618

Adusumalli, H. P. (2018). Digitization in Agriculture: A Timely Challenge for Ecological Perspectives. *Asia Pacific Journal of Energy and Environment*, *5*(2), 97-102. https://doi.org/10.18034/apjee.v5i2.619

Adusumalli, H. P., & Pasupuleti, M. B. (2017). Applications and Practices of Big Data for Development. *Asian Business Review*, *7*(3), 111-116. https://doi.org/10.18034/abr.v7i3.597

Borgia, E. (2014). The Internet of Things vision: Key features, applications and open issues. *Comput. Commun., 54*, 1–31.

Dehury, C. K., Sahoo, P. K. (2016). Design and implementation of a novel service management framework for IoT devices in cloud. *J. Syst. Softw., 119*, 149–161.

Pasupuleti, M. B. (2015a). Data Science: The Sexiest Job in this Century. *International Journal of Reciprocal Symmetry and Physical Sciences*, *2*, 8–11. Retrieved from https://upright.pub/index.php/ijrsps/article/view/56

Pasupuleti, M. B. (2015b). Problems from the Past, Problems from the Future, and Data Science Solutions. *ABC Journal of Advanced Research*, *4*(2), 153-160. https://doi.org/10.18034/abcjar.v4i2.614

Pasupuleti, M. B. (2015c). Stimulating Statistics in the Epoch of Data-Driven Innovations and Data Science. *Asian Journal of Applied Science and Engineering*, *4*, 251–254. Retrieved from https://upright.pub/index.php/ajase/article/view/55

Pasupuleti, M. B. (2016a). The Use of Big Data Analytics in Medical Applications. *Malaysian Journal of Medical and Biological Research*, *3*(2), 111-116. https://doi.org/10.18034/mjmbr.v3i2.615

Pasupuleti, M. B. (2016b). Data Scientist Careers: Applied Orientation for the Beginners. *Global Disclosure of Economics and Business*, *5*(2), 125-132. https://doi.org/10.18034/gdeb.v5i2.617

Pasupuleti, M. B., & Adusumalli, H. P. (2018). Digital Transformation of the High-Technology Manufacturing: An Overview of Main Blockades. *American Journal of Trade and Policy*, *5*(3), 139-142. https://doi.org/10.18034/ajtp.v5i3.599

Ray, P. P. (2016). A survey of IoT cloud platforms. *Future Comput. Inform. J., 1*, 35–46.

Truong, H. L., Dustdar, S. (2015). Principles for engineering IoT cloud systems. *IEEE Cloud Comput.*, *2*, 68–76.

Xia, F., Yang, L.T., Wang, L., Vinel, A. (2012). Internet of things. *Int. J. Commun. Syst., 25*, 1101–1102.

--0--