International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 09 | Sep 2021 www.irjet.net p-ISSN: 2395-0072

# Software-based IoT Framework Generation

**Rajesh Kumar Adla[1], Venkata Mohan Krishna Challa[2]**

[1]*Founder and Director – IoT Null Innovation Inc. Hyderabad, India*
[2]*Venkata Mohan Krishna Challa Engineer-Intern, Null Innovation Inc.*

---

***Abstract* —** 'Internet of Things' is a network of Interconnected objects that can collect and exchange data through the internet. It enables the interconnection of devices, facilitating remote control, and monitoring of activity. Over the years, the importance of IoT has only increased, and the applications that we see today are the proof. Smart cities, smart homes, and smart devices are the buzz that makes IoT, a futuristic and fast-growing technology. IoT adaption is driven by growing internet speeds and upcoming 5G technologies. In the forthcoming days, the usage of IoT appliances will be very high due to the ease of accessibility of high-speed internet. IoT product development involves multi-layer development involving multiple skillsets, and one needs to have expertise to build this multi-layer IoT Stack. It is difficult, time taking and expensive to employ engineers with expertise in each system-level layer to build all five layers of IoT stack. That is where FrameGEN, a software-based IoT framework generator, comes into the picture. FrameGEN creates a framework, including security and compliance guidelines, to help IoT enthusiasts, researchers, and product teams build their own IoT product. In this paper, we will know the working of this software.

***Keywords— Cloud Application, Device Software, Hardware, Internet of Things, IoT Stack, Software as a Service (SaaS).***

## I. INTRODUCTION

IoT has become an integral part of modern-day technology. As the term says, it is the network of objects that can collect and exchange information through the means of the internet. This kind of device monitoring reduces human interaction and improves productivity to a whole new level [5]. The Internet is the body that connects the devices in an IoT appliance together. High-speed internet enables quick response to IoT devices, which makes the usage of IoT devices much more comfortable and efficient. It has also paved the way for remote access to the products. As the speed of the internet will only be increasing in the coming days, the future of IoT is boundless. One of the main shortcomings of the present models is the Framework doesn't include a blueprint of all layers of IoT stack, and IoT enthusiasts are not guided on the quick development of the IoT stack. So IoT developers are having a hard time while building their products [1]. FrameGEN, a software-based framework generator, provides a blueprint that guides the user throughout the product development cycle. The whole process within the FrameGEN happens through five different stages, where each stage represents a layer in the IoT stack. For a given product, FrameGEN will provide a customized framework for every stack layer, in-spite of user limited expertise in building a few layers and no expertise in others. In order to make this framework easy to understand, FrameGEN includes block diagrams, components, flow charts, interfaces and over-all architecture of system [11]

The novelty of this system lies in the component selection and integration that it defines, based on the user inputs for each layer of IoT stack. The Hardware architecture is decided based on the features, for instance, all the sensing elements of the device. Whereas the software drivers, cloud storage are decided based on the internet protocol, data size, and rate of transfer. This novelty of the system can be further enhanced in the future using AI technology.

The paper discusses the general IoT stack layer's division in section-2. Then, FrameGEN and its architecture are described in sections-3,4. Later in section-5, a brief view of queries was given. then the paper concludes in section- 6

### *A. Related Work*

The paper "A Software Framework for Internet of Things," discusses a software which is an animation authoring tool used to generate a program based on state-transition diagrams [10]. This software helps to translate visualization into a code format which will be helpful in making IoT product. However, this Framework offers limited capability, which is further enhanced by FrameGEN. This paper discusses the implementation of FrameGEN and its offerings. FrameGEN generates a framework for each layer of IoT along with security, compliance, and integration guidelines useful to start IoT product development from scratch.

## II. IOT ARCHITECTURE

An IoT product consists of many complex areas of technology. So, understanding and making one is tricky. In order to make the whole process easier, a typical IoT architecture can be divided into five stages/layers [3]. By breaking down the entire procedure into five layers, one can easily make a fully functional IoT product. This

architecture is made based on the field of work involved in a typical IoT device [2,4].

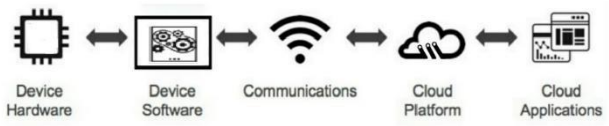


Fig.1 - IoT stack layer

*A. Device Hardware*

The device hardware section acts as the interface between the real and digital worlds. The hardware used in IoT systems includes remote dashboard devices, control devices, and sensors. Hardware is the part that collects the data using sensors. In the device hardware, different types of sensors, like pressure measurement, temperature measurement, accelerometer, etc., can be used to sense various parameters based on the requirement. After collecting data using these sensors, microcontrollers, or microprocessors are used to process the data, and the processed data is made ready to be transmitted to different stack layers [3].

*B. Device Software*

Device software turns the device into a 'smart device.' This part enables a hardware device to serve multiple applications depending on the software it is running. A smart device can be created when Device hardware and software work together. Device software allows the user to implement the communication with the Cloud or other local devices. The user can perform real-time analytics, data acquisition from the device's sensors. Using this a customizable generic hardware can be made that gives your hardware more flexibility down the road. The risks that occur in device hardware can be reduced by using device software [3,8].

*C. Communication*

Communication refers to all the different ways the device will exchange information with the rest of the world. This includes both physical networks and protocols. In this IoT technology stack layer, different types of network communication platforms can be used that will connect the device hardware to the cloud, and then to application. Bluetooth [8]. Wi-Fi and Zigbee are widely used to establish internet connectivity for IoT.

*D. Cloud platform*

The Cloud platform is the backbone of the IoT product. One of the challenges we face while using an IoT application is that they can generate a massive amount of data [3]. To tackle this, a scalable and secure data management solution is a must since the beginning. Data analytics, like machine learning, data compression, perform forecasts, plays a significant role in transmitting data [7,8].

*E. Cloud Application*

Cloud application is the user interface, through which the customer interacts with the device and vice versa. This application is mostly web-based, and a separate app for desktop, mobile, wearables can be used based on the user's needs [3]. Cloud application is the primary source of interaction with IoT products so the user can access the IoT device at any point in time with his smartphone. This enables remote monitoring, controlling, and interaction with the actual device. Since the user can directly access IoT products, it is essential to make application user friendly [8]. Due to the presence of more users in industrial IoT applications, multiple applications have to be used [6].

## III. FRAMEGEN AND ITS ARCHITECTURE

FrameGEN is a SaaS-based platform that helps users in developing IoT products. It facilitates the user with data, tools, skill, guidance, expertise, and support required to make an IoT product. FrameGEN helps in making new product designs with extensive user interaction, knowledge sharing, and a framework for accelerated IoT development coupled with regulatory compliance and security risk guidelines. FrameGEN generates customized product framework based on the user's needs. Once the IoT framework is used to create a blueprint of each layer implementation of the IoT stack, users can start the development cycle. FrameGEN generates a framework covering various stages of product building, including Data, Business, Technology [9]. Also users can select tools required for development, like Analyzer, Diagram, or Estimator necessary for the product.

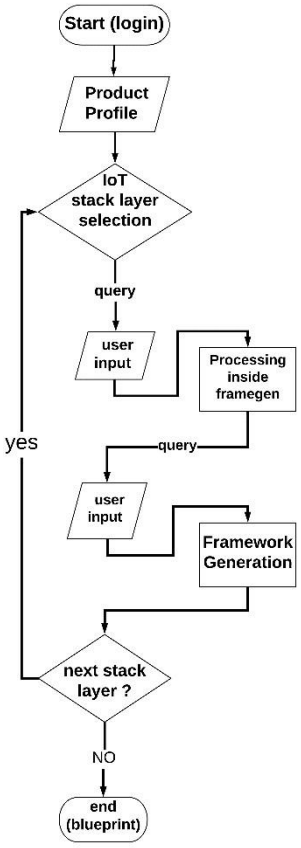


Fig.2 – FrameGEN architecture

The whole process of how a framework is generated by FrameGEN is seen in Fig 2. After logging in, users provide details about the project on the 'Product Profile' page. The details include the project name, description, estimated cost, etc. In the next, page all IoT stack layers required will be listed. By selecting one stack layer, the user can start building the layer, interacting with FrameGEN. These inputs given by the user, for all questions posed by FrameGEN are useful to create the right options, components, and flow for the layer. After the end of the query, FrameGEN creates a framework for the specific layer. This process repeats for all the stack layers, and finally, a product framework will be created. The product framework contains all individual layers, with architecture and flows, to quickly implement. Using this product framework, you can quickly jump onto product development.

## IV. STAGES OF FRAMEWORK GENERATION

The whole process in frameGEN can be divided into three stages, which are initiation, single stack layer development, Product blueprint creation.

### *A. Stage1 - Initiation*

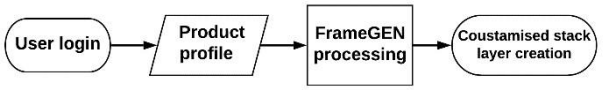


Fig.3-Stage1

In the first stage, the user provides details about the project in the 'Product Profile' page after logging in. The Project Profile page includes project name, description, estimated cost, etc. Based on given inputs, FrameGEN creates a customized stack layer where the user can fill the query and get the Framework.

### *B. Stage 2 – Single Stack Layer Development*

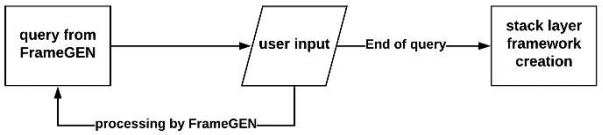


Fig.4- Stage2

This second stage is critical for FrameGEN. A lot of data sharing between users and software takes place in this stage. The user extensively interacts with the software and provides information about the product. User can choose one IoT stack layer among the five and start building the framework for the selected stack layer. After selecting a stack layer, FrameGEN will ask the user to fill a query. This query is about basic details of the product and is generic. For example, in the hardware section, FrameGEN asks about dimensions, weight, environmental conditions, etc. After filing the query, FrameGEN will analyze the inputs and prepare the next set of query, which will be specific to the product. And this process happens multiple times every time the level of query increases when compared to the previous query. At the end of all set of query, a framework for the stack layer will be generated

### *C. Stage 3 – Product Blueprint Creation*

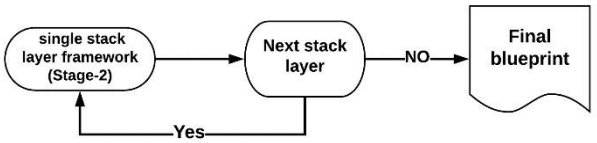


Fig.5- Stage3

In the final stage, the user can select the next stack layer. After selection, stage-2 will be repeated for the selected stack layers. After completing any layer, the user can access the frameworks for all stack layers developed until then. At the end of the final stage i.e. complete development for all stack layers, a complete product blueprint for the IoT product will be generated. Users can also opt for single layer frameworks as per their development needs.

## V. GENERAL QUERIES IN FRAMEGEN

### *A. Hardware*

In this part, FrameGEN enquires the user about hardware specifications. In this section, FrameGEN asks some necessary details like the dimensions and weight of IoT products. Questions generated by frameGEN include the options for development boards like microcontroller, microprocessor, and FPGA. Users can also choose the required sensors in this section. As an example, if one has to develop a product which includes temperature measurement, the user can select a preferable option for temperature measuring sensor in this section. Aside from these questions, FrameGEN also enquires about the user about communication protocol, whether the device is wired or wireless, precautions that need to be taken while using the hardware, the power supply used, etc.

### *B. Software*

This part of the IoT technology stack serves as the glue between the real world (Hardware) and the Cloud Applications. In this section, FrameGEN enquires about the user about the graphical user interface (GUI) used, type of video port implemented, how debugging can be done for the software, software running time (uptime and downtime), the device drivers required. In addition to these questions, FrameGEN also enquires about the user about the type of programming language used, memory management, threats, interrupts, risk management, etc.

### *C. Communication*

Selecting the right communication mechanisms is a vital part of constructing an IoT stack. It will determine not only how the data gets transmitted between the user and Cloud (for example Wi-Fi, WAN, LAN, etc.), but also the communication with third-party devices in the same building or city. In this section, FrameGEN enquires the user about the mode of data communication, speed, bandwidth, power consumption, etc. Based on the given details, the user can select a mode of transmission, communication channel, etc. FrameGEN helps the user with communication protocol, data encryption, risk management, and security.

### *D. Cloud Platform*

Data sharing is usually done by exposing APIs at either the Cloud level or the device level. Cloud APIs allow customers and partners to either interact with the devices or to exchange data. FrameGEN enquires with the user about cloud storage needs. Some of them are the data processing rate, the data intensity of application, data width, available APIs, device management, security management, etc. In this section, some other options that user can choose are the Mode of visualization of data, Data retrieval methods, whether Big data analytics should be used or not, type of analytics library used, etc. Since the Cloud platform is one of the primary layers where security issues arise, the user should be more cautious while choosing a cloud service. So, FrameGEN helps the user while choosing a reliable and affordable cloud platform based on their requirements like data visualization, web interface, etc.

### *E. Cloud application*

This user interface allows the user to access smart devices anytime and from anywhere. While creating an interface, it is essential to understand what a user needs to make a product useful and successful. Cloud application has a vital role while choosing a cost-efficient cloud platform. Some of the necessary details of the product in this layer are target user (mobile, tablet, desktop, web), number of screens required, operating system (android, iOS, Ubuntu, etc.), system requirements for running the app (processor, RAM, ROM), internet connectivity, etc. Based on the above details, FrameGEN provides a framework with the appropriate options. For example, the coding language used, database implementation language, authentication process, etc. The key point here is we need to identify what parameters the customer wants to track, and how does he like them to be presented to him over an app.

## VI. PRODUCT USE CASE

An Innovator wants to develop an IoT product, the "Remote Patient Monitoring" system, to help control the spread of Covid-19. Being a mechanical engineer, he doesn't have much knowledge about IoT Stack and how to develop each layer like Hardware, Software, Communication, Cloud, and Cloud App. His limited software knowledge couldn't be of great help to model the product per customer need. He came to know of FrameGEN and started using it for generating an IoT framework so that he uses it as an engineering blueprint. In the Hardware section of FrameGEN, he entered all features and identified, sensors, controllers, and interfaces required for his product. In the Device Software section, FrameGEN asked him about the firmware features and suggested suitable drivers and controls implement features. In addition, the communication layer section helped the user to choose an internet protocol, data rate, and standards. At

last, using the cloud and cloud applications section, he selected cloud storage, data size, rate of transfer, and finally, a useful app that contains all parameters user loves to track. FrameGEN blueprint was a basis for the innovator to start his product development cycle and achieve product success.

## VII. CONCLUSION

As the Internet of things has become a global sensation, it also provided a promising opportunity to develop crucial industrial research and applications. In the future, there is a high scope for IoT technology. In order to successfully make an IoT product, we need to address, an IoT enthusiast or developer's needs to quickly develop, deliver, and market. The complexity of development, expertise in more than one area of technology, is a bottleneck for accelerated IoT development. FrameGEN empowers IoT developers with guidance, tools, and the Product Framework. Anyone can build an IoT product now, using FrameGEN. The future roadmap for FrameGEN includes an AI engine that enhances user interaction, system architecture, and dynamically guides, users build a product as per the latest standards, security, and compliances.

## VIII. ACKNOWLEDGMENT

We would like to thank IIT Bhubaneswar Electrical Engineering Interns to assist with this research paper. The authors would like to thank the editor and the anonymous reviewers for their constructive and valuable comments. My respect for my teacher Daniel Elizalde for introducing and inspiring me on 'Internet-of-Things' (IoT). Finally, my love for Ram Dhanush, my four-year-old son, for keeping calm while I work.

## REFERENCES

[1]. A. Alvarez, "Microsoft Announces IoT Central: SaaS Platform to simplify the Internet-of-Things", May 2, 2017 [Online]Avalable:https://www.starwindsoftware.com/blog/micr osoft-announces-iot-central-saas-platform-to-simplify-the- internet-of-things Microsoft IoT Architecture, [accessed Sept. 10,2017].

[2]. N. Tyagi, "A REFERENCE ARCHITECTURE For IoT", International Journal of Computer Engineering and Applications, vol. X no.I, Jan.2016.

[3] "IoT Product Managers – A Guide to the IoT Technology Stack" by Daniel Ezdale [online] Available: https://www.iotforall.com/iot-product-managers-guide-iot- technology-stack/[Accessed May 3, 2020]

[4]. Timo Niemirepo, Markus Sihvonen, Vesa Jordan and Juhani Heinilä, "Service Platform for Automated IoT Service Provisioning", 2015 9th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, July 2015

[5]. Er.Pooja Yadav, Er.Ankur Mittal and Hemant Yadav, "IoT: Challenges and Issues in Indian Perspective", 2018 3rd International Conference On Internet of Things: Smart Innovation and Usages (IoT-SIU), Feb 2018.

[6]. Sang Gi Hong, Nae Soo Kim, and Taewook Heo, "A smartphone connected software updating framework for IoT devices", 2015 International Symposium on Consumer Electronics (ISCE), June 2015.

[7]. Abdur Rahim Biswas and Raffaele Giaffreda, "IoT and cloud convergence: Opportunities and challenges", 2014 IEEE World Forum on Internet of Things (WF-IoT), March 2014.

[8]. IoT technology stack explained in detail [online] Available:https://engineering.eckovation.com/iotstack // [Accessed May 3, 2020]

[9]. Framegen, NullInnovation website Available:

https://nulliot.com/framegen%5E$ [Accessed May 3, 2020]

[10]. T. Inui, M. Kohana, S. Okamoto and M. Kamada, "A Software Framework for Internet of Things," 2015 18th International Conference on Network-Based Information Systems, Taipei, 2015, pp. 706-709, doi: 10.1109/NBiS.2015.108.

[11]. A. Perkusich, H. O. Almeida and D. H. de Araujo, "A software framework for real-time embedded automation and control systems," EFTA 2003. 2003 IEEE Conference on Emerging Technologies and Factory Automation. Proceedings (Cat. No.03TH8696), Lisbon, Portugal, 2003, pp. 181-184 vol.2, doi: 10.1109/ETFA.2003.1248694.