Towards a Dynamic Scalable IoT Computing Platform Architecture

Desoky Abdelqawy[1], Amr Kamel[2], Soha Makady[3]
Faculty of Computers and Artificial Intelligence
Cairo University, Egypt

***Abstract*—Internet of Things (IoT) has become an interesting topic among technology titans and different business groups. IoT platforms have been introduced to support the development of IoT applications and services. Such platforms connect the real and virtual worlds of objects, systems and people. Even though IoT platforms increasingly target various domains, they still suffer from various limitations. (1) Integrating hardware devices from different providers/vendors (thereafter referenced as heterogeneous hardware) is still a subtle task. (2) Providing a scalable solution without altering the end user privacy (e.g., through the use of cloud platforms) is hard to achieve. (3) Handling IoT Applications reliability as well as platform reliability is still not fully supported. (4) Addressing Safety-critical applications needs are still not covered by such platforms. A novel scalable dynamic computing platform architecture is proposed to address such limitations and provide simultaneous support for five non-functional requirements. The supported non-functional requirements are scalability, reliability, privacy, timing for real-time systems and safety. The proposed architecture uses a novel network topology design, virtualization and containerization concepts, along with a service-oriented architecture. We present and use a smart home case study to evaluate how traditional IoT platform architectures are compared to the proposed architecture, in terms of supporting the five non-functional requirements.**

***Keywords*—Interent of Things (IoT); IoT platforms; IoT architecture; edge computing**

I. Introduction

The world is currently changing very fast, jogging to be Smart. Smart Cities [1], Smart Homes [2] and Smart Factories [3] and Smart Grid [4] are bright terms the world is currently looking up to. Internet Of Things [IoT] technology is considered the main player to achieve such aspiration; Gartner [5] reported that by 2020, 95% of new product designs will contain IoT Technology. IoT had been included in the list of six "Disruptive Civil Technologies" with potential impact on US national power by the US National Intelligence Council [6]. In [7], There will be 50 billion things connected to the internet by 2020 as predicted by Cisco Internet Business Solutions Group.

IoT is defined as a network of devices/things coupled with sensors, actuators, software as well as required electronics to make them able to collect, process and share data. From another perspective an IoT is an architectural framework that permits the integration and/or data exchange between the physical world and computer systems through the underlying network infrastructure. This network is orchestrated with what so called an IoT platform. An IoT platform is the key software component that facilitates the development of scalable IoT applications and services that connect the real and virtual worlds between objects, systems and people. As described in [8], such platforms have to meet the expectation of different players in the IoT ecosystem. i.e (1) Device vendors require a standardized communication protocol for seamless integration and operation (2) Application developers need a simplified development support to focus on application development instead of integration and deployment issues. (3) The providers of platforms and related services seek a clean and simplified way to extend and support their services. (4) The end-users demand security and privacy support.

More than one hundred of such platforms have been created over previous years [9]. Such platforms come in various shapes, and sizes. Yet, there is still a lack of any defined agreement or a standard to manage such technology (e.g., a standard communication protocol, a standard architecture and deployment methodologies, a defined and dedicated market place. [10].

Therefore, various studies have been conducted in [8], [11], [12] and [10] to evaluate IoT platforms landscape, existing IoT architectures, and assess whether such platforms satisfy the IoT ecosystem needs. Such studies concluded that although existing IoT platforms cover a wide-range requirements for IoT platforms, the following four non-functional requirement still remain relatively unexplored: (1) system-wide scalable dynamic resource discovery, (2) reliability (3) Real Time support and (4) privacy.

In this paper, we propose a scalable computing platform architecture, that simultaneously satisfies the above mentioned four non-functional requirements in addition to securing the required support for safety critical IoT Applications.

The remainder of this paper is structured as follows. Section II provides a background on traditional IoT platform architectures. Section III presents a motivational scenario for an extendable temporary virtual key management system as a Smart-Home use-case. In Section IV we present our proposed architecture and apply it to the proposed smart-home use-case in Section V. We compare our proposed architecture against traditional IoT architectures in Section VI. Section VII presents the related work, whereas Section VIII concludes the paper.

II. Background

Simply IoT platform could be defined as the enabler platform addressing IoT Full stack. Such IoT stack includes devices/actions/connectivity management, analytic, developer ecosystem, orchestration and open-external interfaces [9].

A classification for IoT Platforms could be done from platform architecture point of view as defined in [8].

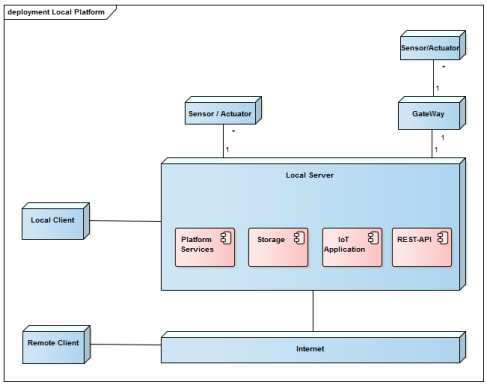


Fig. 1. UML deployment Diagram For Local Based Architecture IoT Platforms

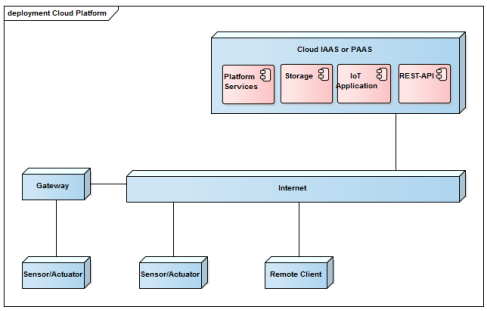


Fig. 2. UML deployment Diagram For Cloud Based Architecture IoT Platforms

- Local Platform
  Figure 1 models that type of platform, where there is a local server (i.e., a centralized node) that acts as a single computing resource. Such node contains four main components: the platform's services, the data storage, the hosted IoT applications, and the REST API to expose the platform services through Internet for remote clients. The local server connects to sensors and actuators through either a wired or a wireless connection. The connection could be done in a direct manner or through a gateway that acts as a bridge for such a group of sensors or actuators to the platform. This centralized node might expose a REST full API to Remote Client through Internet i.e mobile applications or web-app dashboards. Alternatively a local client can access the platform functionalities through a direct connection with the local server.
- Cloud Platform
  Figure 2 models that type of platform, where the sensors and/or actuators are connected directly through the internet, or through a gateway, to the cloud. The Cloud provides the required services, storage and needed computing resource for the applications in either infrastructure-as-a-service (IAAS) or platform-as-a-service (PAAS). Remote Client can access the platform functionalities through an Internet connection.

A summary for a set of non-functional requirements for IoT platforms has been presented in [13] that includes the following: (1) Scalability; an IoT platform shall support expansion with heterogeneous devices and applications diversity inside ultra large network. (2) Reliability, an IoT platform shall have the capabilities to cope with the IoT nodes/devices constrained resources and the dynamic nature of IoT hubs/networks where devices/nodes not always available all the time. (3) Timing for real-time applications, IoT platform shall be able to serve real time applications with timing requirements. (4) Safety, Where IoT platform shall provide the required support to execute safety critical applications i.e. redundancy support, and application migration to recover from hardware failures.

TABLE I. SUMMARY OF PLATFORMS ARCHITECTURE PROPERTIES

Architecture / Properties	Local Platforms	Cloud Platforms
Scalability	✗	✓
Reliability	✗	✓
Privacy	✓	✗
Timing for Real-Time	✓	✗
Safety	✗	✓

Table I presents an analysis for whether existing IoT platforms architecture (mainly local and cloud based platforms) support the above mentioned non functional requirements. we further explain as follows:

- Scalability
  Cloud based platforms are scalable by nature where computing capabilities can be scaled up with pay-as-you-go model. On the other hand, local based platforms suffer from fixed computing capabilities which is defined from the beginning thus limiting their scalability. In case of un-reliable connections with the cloud, Local based platforms could be considered as more reliable.
- Reliability
  In case of a reliable connection, cloud based platforms could be considered more reliable due to high availability of resources and infrastructure reliability. On the other hand Local-based platform might not provide the needed support for reliability due it's fixed static resources available from the beginning.
- Privacy
  Local based platforms satisfy privacy through hosting all the data of an IoT application locally. Such local hosting gives the user full control on who could be authorized to access such data. On the contrary, cloud-based platforms cannot satisfy privacy as the data is hosted remotely on a cloud. Such hosting raises issues specially when there is no clear strategy for data ownership.
- Timing for Real-Time
  Local based platforms provide the needed support for Real-Time application since it does not suffer from latency related issues which is part of cloud based platforms by nature.
- Safety
  Safety Critical applications are defined as applications where failure might lead to death or serious

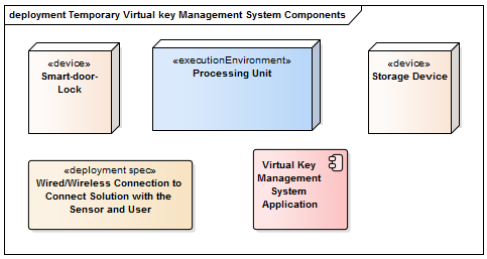


Fig. 3. Use-case main components

injury to people, severe damage in equipments or environment harm. This type of applications exist heavily in Smart Factories and Smart-Vehicles. Those applications might need a special type of platforms that support redundancy, cross checks, freedom from interference. Cloud based platforms have the available resources to support redundancies for such applications. On the other hand, local-based platforms cannot easily be extended to provide the required redundant copies. That is mainly because of the fixed resources defined from the beginning within local-based platforms.

III. Motivational Scenario

Smart-Home is a home with an automation system that enables electronic, electrical and technology based tasks within a home. A home Automation system might control lighting, climates, entertainment systems. A home automation system could also manage home security such as access control and alarm systems.

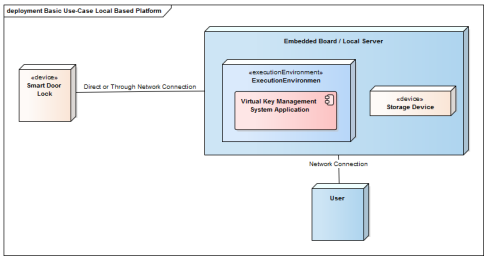
A. Temporary Virtual Key System

An owner of a smart house decides to rent his home in the summer to different tenants every month. Accordingly, different tenants could be using the house with the need to share the physical key of the house across such tenants. Any house tenant could miss closing the door of the house by mistake, hence increasing the chances of robbing such house. Accordingly, a virtual key management system would be a highly needed feature in such a smart house. Such system would allow the house owner to create and share a virtually time bounded house entry key.

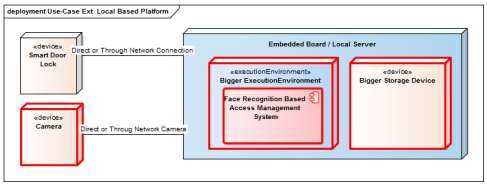
Figure 3 shows the main components needed to setup and deploy such system. Those components include: a smart door access lock (a device actuator), a virtual key management software system/application (IoT Application), processing unit (Execution environment) to execute the application and storage device to persist the application.

IoT platform is responsible for managing and organizing such components together to achieve use-case requirements. Local or Cloud based IoT platforms could be utilized to deploy the above mentioned simple concrete use-case.

A local based platform architecture for the smart home virtual key management system presented in Figure 4a has a single node that consists of a storage to store virtual key management software application, processing unit to execute



(a) Use-case Ext-1: Local based Platform



(b) Case II

Fig. 4. Local Based Platform Use-case(s) Architecture

it and wired/wireless connection with the smart door lock actuator device. Such architecture is defined since the initial design of such Figure 5a presents an alternative cloud-based platform architecture for the same smart home virtual key management system where the smart door lock is connected directly to a cloud hosted virtual key management software application. Such application would be stored and executed inside cloud infrastructure and the control signal is pushed back to the smart lock over Internet. Also such architecture is considered a scalable one due to the ease of adding more processing power, it suffers from privacy and reliability issues in-case of un-stable Internet connection.

B. Supporting Face-recognition within Smart Home Virtual Key Management System

The house/property owner decides to extend the basic use-case through the addition of a door mounted camera to enable face-recognition capabilities over the provided temporary virtual key access solution. Such extension would help the house owner to validate if the person who is trying to enter the house is from an allowed list of faces, and additionally owns a correct virtual key. Such an extension would demand installing an en-adding more computational resources to support the facial recognition application needed to identify.

For the local based platform shown in 4a, such an extension would demand more computational resources to support facial recognition application of the camera. As the architecture is a preset local one, the owner would end up replacing the platform with a totally new one that supports the needed camera setup.

Figure 4b presents architecture modification for such extension where a new node with bigger capabilities has replaced the original node used to cover basic use-case and the camera has been connected to the platform beside existing smart door lock. Such node will include a newly developed or extended software application to realize face recognition based access.

Cloud Based platform might support such extension if the

mounted camera has the required capabilities to be directly connected to the cloud. Figure 5b presents the required architecture change through adding such Internet connected Camera and re-implement/extending the existing cloud hosted/stored temporary virtual key access application to receive the camera feed and support face-recognition based access. The cloud cost package used to cover the basic use-case might need to be upgraded. Since camera feed still needs to be sent for processing on the server, that solution suffers from latency issues. Camera Feed now is accessed/owned by the cloud provider thus leading to privacy issues as anyone could easily access that stream of data. Furthermore, in-case of a limited bandwidth situation this solution might not be applicable all together where the limited bandwidth might affect camera feed transmission to the cloud for recognition.

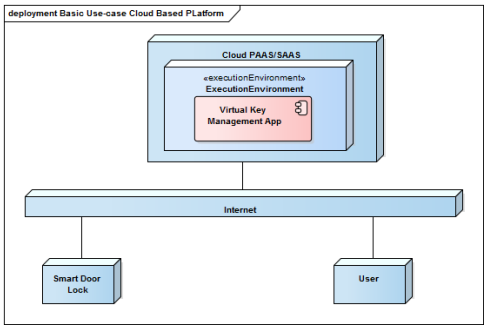
C. Supporting Window Status Monitoring within Smart Home Virtual Key Management System

On the other hand if the house/property owner decides to further extend the smart home through adding a door/window status sensors. Such sensors might be used to detect if door/window status is opened or closed, so that he can monitor/confirm his hours/property status; he will suffer from the same limitation described in first extension. Another possible use case extension is to add a door/window status monitoring application. Such application kind is considered a safety critical application assume that the smart house's door monitoring system crashes suddenly while the house owner is away from the house. With only one instance of the monitoring system, a thief would easily access the house/property. Accordingly two instances of the monitoring system need to be present. Such two instances imply un-needed extra cost in both cloud and local based solutions.

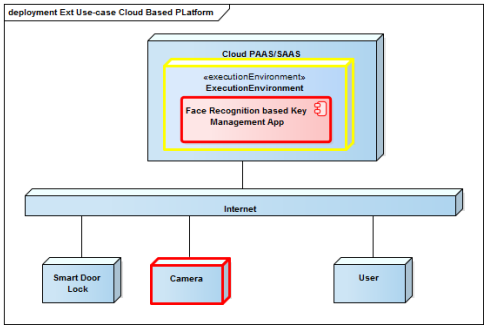
Although there exists an under-utilized processing power that not used all the time for example facial recognition door access processing resources is only used during door accessing process. Home automation systems and IoT applications in general are scalable, dynamic and heterogeneous by nature. Use-case(s) extensions are infinite. There is a high need for dynamic, scalable platform/middle-ware that absorb and support such nature. In the next section we will present a proposed dynamic scalable computing platform architecture for IoT hub(s). After that we will evaluate it against the described use-case with its extensions.

IV. Proposed Architecture

The following section will present a dynamic scalable computing IoT platform architecture used to manage IoT networks/Sub-networks resources. Main objectives for such architecture are to (i) Secure the required computing resources for different IoT applications without breaching users' privacy. Furthermore, such computing resources should not lead to unneeded latency, specialty within applications with real-time demands. (ii) Abstract the underline IoT network infrastructure for such IoT applications. Through such abstraction IoT applications will be totally decoupled from underline hardware constrains as well as sensors/actuators availability and providers. (iii) Orchestrate the network and (IV) Maximize resource utilization.



(a) Use-case Architecture Utilizing Cloud based Platform



(b) Use-case Ext-1: Use-case Cloud based Platform

Fig. 5. Cloud Based Platform Use-case(s) Architecture

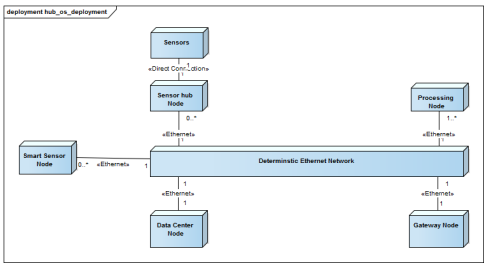


Fig. 6. Proposed Platform Architecture Network Topology.

A. Architecture Overview

To achieve the proposed architecture two main objectives will be introduced:

- Topology Change:
  Hub/Subnetwork topology need to be introduced
- Software Architecture Change:
  Extending service oriented architecture concepts to bring cloud flexibility into the local based platforms.

Topology changes will be discussed in section IV-A1, and Software architecture changes will be presented in section IV-A2.

1) Platform Hardware Network Topology: Figure 6 illustrate the proposed architecture Network topology which contains the following:

- Data Center Node:
  A standalone device/machine used to host the platform

system services and store the available IoT applications package.

- Processing Node(s):
  A one or more standalone device(s)/machine(s) used to host IoT application while executing. These nodes could be added or removed dynamically and the platform will adapt itself and the running application against such situations.
- Gateway Node:
  A standalone device/machine used to act as a translation unit that bridge the communication between IoT Hub/sub-network currently managed by the platform and external world.
- Sensors Hub Node:
  A standalone device/machine used to group number of different or similar sensors to be exposed for IoT application(s). This node is optional
- Smart Sensor(s) Node:
  A standalone sensor that directly connected to IoT Hub/Sub-network.
- Time Sensitive Networking:
  A network medium that support IEEE TSN which has clock synchronization profile 802.1AS based on 1588v2 and messages are forwarded as part of scheduled queues 802.1Qbv.

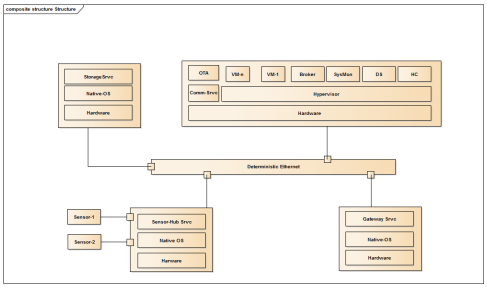


Fig. 7. Proposed Platform Software Architecture.

2) Platform Software Architecture: Figure 7 illustrate the proposed platform architecture software system services within IoT hub network which includes the following services:

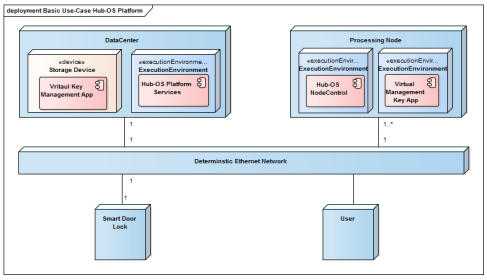
- Data Storage Service:
  A service to store IoT application(s) packages in a storage efficient manner while providing them on-demand over network connection. IoT application package should contains (1) IoT application binary image that hold the application executable as well as all its dependency in a container or Virtual Machine image format. (2) Application manifest file which contains all requirement to be provided by the platform i.e. maximum needed cpu load, required specific architecture (X86 or Arm) existing of acceleration, memory needs..etc.
- System Monitor Service:
  A service used to monitor the overall platform status including which IoT application is running over which processing node, loads of each processing node as well as the availability of one or more sensor(s) or services.
- Broker Service:
  A service that manage IoT application(s) scheduling [activation and shutdown] over the clustered processing nodes. This service abstracts the scheduling algorithms to maximize the resource utilization through using all available processing nodes or minimize the power usage through packing all application VMs to as minimum as required processing nodes.
- Node-Control Service:
  A service used to manage the underneath processing node as a generic computing unit, it will dynamically configure, load and execute an IoT application sent through network by Broker Service and gather real-time statistics information about it and send it to System Monitor Service.
- Sensor-As-a-Service Service:
  A service that abstract the underneath sensors/actuator and facilitate its discovery and usage by the platform. this service might be part of Sensors/actuators Hub Node or smart sensor/actuator node.
- Gateway Service:
  A service used to bridge the platform specific Ethernet communication to other external different Networks or Internet.
- OTA Service:
  A service used to manage IoT application packages versions and control their updates inside data center node storage.

V. Applying Proposed Architecture To Temporary Virtual Key System

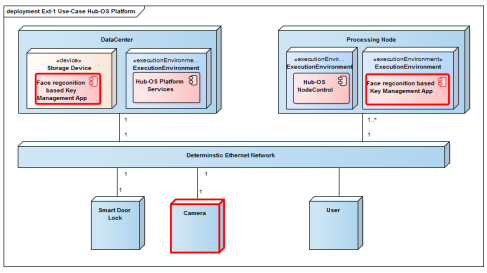
Referring to our motivational scenario presented in Section III we will apply our proposed architecture to alive-ate the limitations of both local and cloud based IoT platforms.

Figure 8a presents basic use-case architecture based on the introduced platform architecture. Data Center Node will store temporary virtual key management system application package. This package contains (i) virtual key management software executable binary container image, (ii) a resource configuration file that describes the required resources what is the maximum amount of memory and CPU resource needed, the required access to certain sensor(s) or actuators and (iii) an optional Application activation binary which is a standalone executable that interact with the architecture platform Sdk to Control the application execution based on certain condition i.e Existing of a sensor output of another application. Data Center Node will host also system services described in section IV. Processing Node which contains a Node Control system service. Such service will start/terminate/monitor the temporary virtual key management software application in separate Execution environment based on need. Smart Door Lock actuator that is controlled by application executed inside processing node.

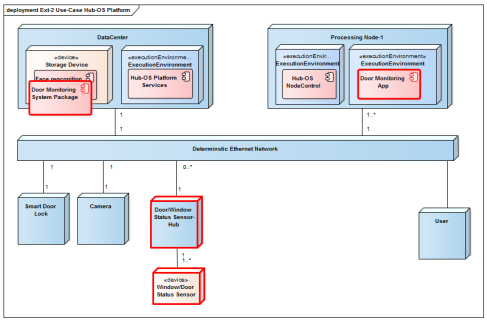
Scalability is one of the main features of the presented platform architecture. To support Use-case Extension for face



(a) Use-case Architecture Utilizing The proposed Platform



(b) Use-case Ext-1: Proposed Platform Architecture



(c) Use-case Ext-2: Proposed Platform Architecture

Fig. 8. Proposed Platform Use-case(s) Architecture

recognition based virtual key management system we will need to connect the Camera to platform network and update the software to process the camera feed and implement the required application logic. Such application will be hosted/executed on the same processing node used to host/execute the basic use-case application. Figure 8b present such required extension. Camera Feed is locally processed inside local processing node so it's not suffer from neither latency or privacy issues.

To support use-case extension-2 that include door and window monitoring system; The proposed architecture sensor hub node will be added to encapsulate/abstract different number of door/window status sensors. Monitoring Application package will be stored on Data-Center node. Processing Node will host/execute the monitoring application in a separate execution environment. Figure 8c presents such modified architecture. Since door and Window monitoring system is a consider a safety critical application which need redundant execution; The Proposed Architecture easily support such safety related requirement either by allowing for executing two different instance(s) of the application inside a single processing node or into two different nodes.

Unlike the previous platforms architecture (Local and Cloud based), the proposed one provides a scalable, reliable solution with supportive capabilities for real-time and safety critical applications in fully controlled private environment.

VI. Proposed Architecture Comparison against other Architectures

Referring to non-functional requirement discussed in Section II; we will compare our proposed approach against each one of them. Table II show a comparison between the proposed architecture approach and existing Local/Cloud based IoT platforms.

- Scalability:
  The proposed Architecture is build from the ground-up to support be scalable in both Hardware and Software. To extends processing resource capabilities of the platform a Processing node will be attached to the platform network and it will be dynamically discovered by SysMon system service and be available to Broker system service to host application(s).
- Reliability:
  The proposed Architecture separates software storage-node from execution i.e. application package is stored inside a centralized data center node and executed on another processing node based on its availability. With support of SysMon system service as a global monitoring system for the platform; a failure in an application could be easily detected and re-executed. Even in case of Hardware failure the application still could be scheduled to be executed in one of the other available processing nodes.
- Privacy:
  The proposed Architecture is a hub/sub-network manager and orchestrator where every-thing is hosted and executed locally with full data owner-ship and control.
- Timing for Real-Time Support:
  The proposed Architecture utilize Time Sensitive network (TSN) to guarantee latency between nodes so it's provide the required system level support for applications that needs a real-time feature.
- Safety Support:
  Based on full flexibility to execute multiple instance(s) from an application either on the same processing node or on a different ones, the proposed architecture secure the required redundancy at minimum cost.

VII. Related Work

Current advance in IoT researches shows a lot of platforms developments. Large number of these platforms has been surveyed in [8] [10] [11] [12] [13], [14], [15] and [16] concluded the lack of a dynamic scale-able IoT platform that helps in utilizing global system resources, support discovery and composition, reliability, security and privacy; the proposed architecture addresses such features.

Non-functional requirements of IoT platform architectures has been explored separately in the literature. Security and privacy has been addressed in [17], [18] and [19]. In [17] an access control provider (ACP) based solution for has

TABLE II. PROPOSED PLATFORM ARCHITECTURE VS LOCAL AND CLOUD BASED PLATFORMS

Properties \ Architecture	Local Platforms	Cloud Platforms	Proposed Platform Architecture
Scalability	✗	✓	✓
Reliability	✗	✓	✓
Privacy	✓	✗	✓
Timing for Real-Time	✓	✗	✓
Safety	✗	✓	✓

been introduced to support security and privacy requirements of interoperable IoT architecture without any pre-established secret information. In [18] a cooperative system between internet service provider (ISP) and and home-gateway has been introduced to provide efficient yet privacy-aware IoT security services. In [19] the effect of using for-oriented architecture could be used for improving the user-privacy and a mapping of privacy patterns to IoT fog/ cloud architecture has been introduced. Conceptually these work complements the proposed architecture which secure the needed resources to realize and implement such techniques.

Where, Real-Time support has been address in [20], [21] and [22]. In [20] a design for building evacuation as a real-time emergency safety critical IoT application where real-time performance and evacuation time are critical. The proposed architecture easily support such kind of use-case implementation through securing the needed resources to provide a collaborative distributed approach for such applications. In [21] a network optimization techniques has been surveyed as one of the enabler technologies to support real-time application in IoT platforms. they complements the proposed architecture. In [22] IoT Fog computing architecture has been used to leverage user-centric technologies that bring the IoT control and analytics closer to the user and cover latency and real-time support gap in cloud based IoT platform solutions. a fog sensing concepts has been introduced and their major challenges has been analyzed. an IoT-in-the-Fog controller has been introduced that used to probe local resources and manage communication directly with local fog-mediators.

Moreover, Scalability and dynamic nature supports of IoT systems has been addressed in [23] and [24]. In [23] a software defined IoT units concepts has been introduced to encapsulate a fine-grained IoT resources and capabilities. it automate the configuration and provisioning of IoT application in IoT cloud systems. In [24] UBIWARE, LinkSmart, OpenIOT and CHOReOS IoT middle-wares has been analyzed with respect to Scalability and heterogeneity of dynamic IoT environment and they concluded none of these middle-wares/platforms support fully autonomnus and scalable service registration, discovery and composition. as well as no one of them scales well in service discovery and service composition response time.

On the other hand, Service Oriented architecture (SOA) has been used by literature to address IoT challenges such as interoperability as well as Middle-ware design and implementations [25], [26] and [27].

In [25] a Service Oriented architecture for Home Area Network (SoHAN) has been introduced to facilitate and abstract a network of sensors and/or actuators for application developer over 5G networks. they based on a Sensor node that has a processing capabilities to process the sensor data and send it to a gateway hosted in Home premises which send it to the Cloud/server for processing. this architecture is not scale well still suffer from privacy, and scalability issues compared to the proposed architecture. In [26] SOA has been revisited to address the scale, dynamic and heterogeneity of IoT. they introduce probabilistic approach for service discovery to filter out redundant data and support ultra-large number of things. service composition aggregate data stream within a network to reduce the network load. advice eVolution Service Bus (VSB) to enable interconnection of things that adhere to different interaction style through utilizing set of Binding components (BC) to proxy the sensor specific communication protocol to VSB. the porposed architecture align with such refinement through the system services including for ex sensor as a service (SAS) component which is used to bridge/proxy the sensor communication protocol to the proposed architecture. In [27] Network server as a service has been implemented to enable porting Long Range device (LoRa) networks to the cloud. a LoRaWare is a service oriented architecture that allow the developers to enhance the capabilities of LoRa enabled application has been introduced as well. the main focus was interoperability and exposing the Long Range devices.

VIII. CONCLUSION

A scalable computing IoT platform architecture has been introduced in this paper. The Main objectives of such platform architecture are to (i) Secure the required computing resources for different IoT applications, (ii) Abstract the underline IoT network infrastructure for them, (iii) Orchestrate the network and (IV) Maximize resource utilization. A Smart Home application use-case has been introduced for Virtual Key management system with two extensions (I) Face recognition based authentication virtual key management system (II) Door/Window status monitoring system. An evaluation of the proposed architecture to support such application has been introduced compared to local and cloud based platforms.

REFERENCES

[1] H. Schaffers, N. Komninos, M. Pallot, B. Trousse, M. Nilsson, and A. Oliveira, "Smart cities and the future internet: Towards cooperation frameworks for open innovation," in *The Future Internet*, ser. Lecture Notes in Computer Science, J. Domingue, A. Galis, A. Gavras, T. Zahariadis, D. Lambert, F. Cleary, P. Daras, S. Krco, H. Müller, M.-S. Li, H. Schaffers, V. Lotz, F. Alvarez, B. Stiller, S. Karnouskos, S. Avessta, and M. Nilsson, Eds. Springer Berlin Heidelberg, pp. 431–446.

[2] V. Ricquebourg, D. Menga, D. Durand, B. Marhic, L. Delahoche, and C. Logé, "The smart home concept : our immediate future," pp. 23–28.

[3] C. Constantinescu, D. Lucke, and E. Westkämper, "Smart factory - a step towards the next generation of manufacturing."

[4] H. Farhangi, "The path of the smart grid," vol. 8, no. 1, pp. 18–28. [Online]. Available: http://ieeexplore.ieee.org/document/5357331/

[5] Gartner top strategic predictions for 2018 and beyond. [Online]. Available: https://www.gartner.com/smarterwithgartner/gartner-top-strategic-predictions-for-2018-and-beyond/

[6] "Six technologies with potential impacts on US interests out to 2025," p. 48.

[7] D. Evans, "The internet of things: How the next evolution of the internet is changing everything," *CISCO white paper*, vol. 1, no. 2011, pp. 1–11, 2011.

[8] J. Mineraud, O. Mazhelis, X. Su, and S. Tarkoma, "A gap analysis of internet-of-things platforms," vol. 89-90, pp. 5–16. [Online]. Available: http://arxiv.org/abs/1502.01181

[9] A. Sabella, R. Irons-Mclean, and M. Yannuzzi, *Orchestrating and Automating Security for the Internet of Things: Delivering Advanced Security Capabilities from Edge to Cloud for IoT*. Cisco Press, 2018.

[10] IoT cloud platform landscape | 2019 vendor list. [Online]. Available: https://www.postscapes.com/internet-of-things-platforms/

[11] J. Guth, U. Breitenbücher, M. Falkenthal, P. Fremantle, O. Kopp, F. Leymann, and L. Reinfurt, "A detailed analysis of iot platform architectures: concepts, similarities, and differences," in *Internet of Everything*. Springer, 2018, pp. 81–101.

[12] J. Guth, U. Breitenbucher, M. Falkenthal, F. Leymann, and L. Reinfurt, "Comparison of IoT platform architectures: A field study based on a reference architecture," in *2016 Cloudification of the Internet of Things (CIoT)*. IEEE, pp. 1–6. [Online]. Available: http://ieeexplore.ieee.org/document/7872918/

[13] Middleware for internet of things: A survey - IEEE journals & magazine. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7322178/

[14] H. Hejazi, H. Rajab, T. Cinkler, and L. Lengyel, "Survey of platforms for massive IoT," in *2018 IEEE International Conference on Future IoT Technologies (Future IoT)*, pp. 1–8.

[15] L. Bittencourt, R. Immich, R. Sakellariou, N. Fonseca, E. Madeira, M. Curado, L. Villas, L. DaSilva, C. Lee, and O. Rana, "The internet of things, fog and cloud continuum: Integration and challenges," vol. 3-4, pp. 134–155. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2542660518300635

[16] V. Bastidas, M. Helfert, and M. Bezbradica, "A requirements framework for the design of smart city reference architectures," in *Proceedings of the 51st Hawaii International Conference on System Sciences*, 2018.

[17] N. Fotiou and G. C. Polyzos, "Authentication and authorization for interoperable iot architectures," in *International Workshop on Emerging Technologies for Authorization and Authentication*. Springer, 2018, pp. 3–16.

[18] H. Haddadi, V. Christophides, R. Teixeira, K. Cho, S. Suzuki, and A. Perrig, "Siotome: An edge-isp collaborative architecture for iot security," in *Proc. IoTSec*, 2018.

[19] S. Pape and K. Rannenberg, "Applying privacy patterns to the internet of things'(iot) architecture," *Mobile Networks and Applications*, vol. 24, no. 3, pp. 925–933, 2019.

[20] C. Arbib, D. Arcelli, J. Dugdale, M. Moghaddam, and H. Muccini, "Real-time emergency response through performant iot architectures," in *International Conference on Information Systems for Crisis Response and Management (ISCRAM)*, 2019.

[21] N. Srinidhi, S. D. Kumar, and K. Venugopal, "Network optimizations in the internet of things: A review," *Engineering Science and Technology, an International Journal*, vol. 22, no. 1, pp. 1–21, 2019.

[22] S. M. Oteafy and H. S. Hassanein, "Iot in the fog: A roadmap for data-centric iot development," *IEEE Communications Magazine*, vol. 56, no. 3, pp. 157–163, 2018.

[23] S. Nastic, S. Sehic, D.-H. Le, H.-L. Truong, and S. Dustdar, "Provisioning software-defined iot cloud systems," in *2014 international conference on future internet of things and cloud*. IEEE, 2014, pp. 288–295.

[24] A. Palade, C. Cabrera, F. Li, G. White, M. A. Razzaque, and S. Clarke, "Middleware for internet of things: an evaluation in a small-scale iot environment," *Journal of Reliable Intelligent Environments*, vol. 4, no. 1, pp. 3–23, 2018.

[25] M. R. Abd Rahim, R. A. Rashid, A. M. Rateb, M. A. Sarijari, A. S. Abdullah, A. H. F. A. Hamid, H. Sayuti, and N. Fisal, "Service-oriented architecture for iot home area networking in 5 g," *5G Networks: Fundamental Requirements, Enabling Technologies, and Operations Management*, pp. 577–602, 2018.

[26] V. Issarny, G. Bouloukakis, N. Georgantas, and B. Billet, "Revisiting service-oriented architecture for the iot: a middleware perspective," in *International conference on service-oriented computing*. Springer, 2016, pp. 3–17.

[27] K. Tsakos and E. G. Petrakis, "Service oriented architecture for interconnecting lora devices with the cloud," in *International Conference on Advanced Information Networking and Applications*. Springer, 2019, pp. 1082–1093.