iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** XI **Month of publication:** November 2024

DOI: https://doi.org/10.22214/ijraset.2024.65442

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue XI Nov 2024- Available at www.ijraset.com

Deployment and Scalability of Ubuntu Core in IoT Ecosystems

Siddhesh Narnavre[1], Pranav Patil[2], Vedant Surkar[3], Sachin Pandarkar[4], Minal Deshmukh[5]
Electronics and Telecommunications Department, Vishwakarma Institute of Information Technology, Pune, Maharashtra, India

Abstract: ***While the count of IoT devices is increasing every day, so do the incidents associating with IoT devices. Most of the IoT devices lack security protections, thereby posing a threat to them and exposing it to attacks like the one carried out by 2016's Mirai Botnet. Researchers aimed for improving security and mitigating risks associated with IoT devices. To this, the researchers developed various open-source operating systems, as well as dependable and secure operating systems in particular for Internet of Things (IoT) devices. This paper describes the deployment and scalability of one of such operating systems, namely Ubuntu Core.***

I. INTRODUCTION

A. Background

The different abbreviation for the IoT is ”Internet of Things.” To be precise, IoT is all about ”stuff.” IoT equipment is the combination of real-life stuff that is mostly called ”things.” Appliance chipsets, software programs, and other technology solutions all developed and embedded in these are what make them able to connect. Internet of Things is a system that allows different devices and systems to communicate amongst themselves as explained in [1] But it is simpler to say that IoT means technology that connects devices that are found in everyday objects and thus the internet to collect, transmit, and share data. The first and the updated versions of the digital objects are the connected devices named the smart objects. [2] They can come in many shapes and forms, from a simple smart thermostat, for example, to commercial equipment of the highest technology.

All devices of IoT communicate and transfer data through the varieties of communication protocols, such as I2C, UART, SPI, etc. Other devices, gateways, or cloud systems. It can be done over Wi-Fi, Bluetooth, Cellular network (4G/5G), LoRaWAN, Zigbee, etc. Once it collects the data, it sends this data to either a local gateway, cloud server, or edge device for processing. There are basically two kinds of processing architectures for processing data, that is Edge Computing and Cloud Computing. The above terms actually refer to where and how the processing, storing, and analyzing of data take place in a computing environment, especially in the context of IoT and distributed systems. [3]

The vulnerabilities of IoT are mainly due to the integration of so many connected devices and the difficulties in pro-

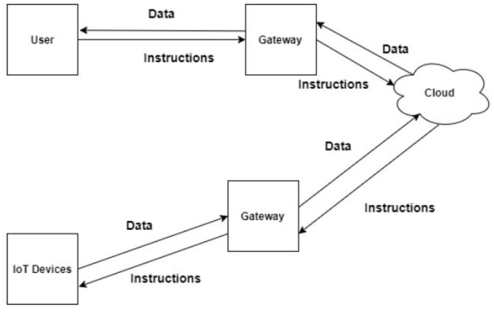


Fig. 1. Data flow and instruction flow of IoT system. [4]

tecting them. Some pervasive threats like Data privacy and security, Insecure communication, device hijacking, lack of standardized security, weak authentication, physical security risks, software vulnerabilities, denial of service attacks, supply chain attacks and legal and compliance issues. Therefore, IoT constitutes a massive problem of privacy and security caused by weakly authentication, the loss of sensitive data, insecure communication and the absence of unified security practices that is the main target for hackers and other malicious entities as explained in [2].

©IJRASET: All Rights are Reserved

B. Problem Statement

These security technologies develop very crucially, as IoT devices are prone to diverse threats, including data breaches, unauthorized access, hijacking, and even physical tampering. [1] The billions of connected devices handling sensitive information and performing key tasks make their security extremely significant in order to have the system prevented from cyberattacks, ensure user privacy, and ensure the reliability of the IoT systems in question. Hardware-based security solutions, such as TPMs, secure microcontrollers, and encryption chips are then considered to offer more protection against such attacks but having also unique limitations in low-power consumption and less processing capabilities available. [2]

Several technologies were then taken into design and development to make Iot networks safer and overcome threats caused. These technologies are mainly focused on protecting the data, device authentication, securing communication, and managing a secure enormous number of IoT devices. This may happen by using some of the technologies such as Encryption, Public Key Infrastructure (PKI), Blockchain, Lightweight Cryptography, Secure Boot, Intrusion Detection System (IDS), IoT device management platforms, Network Segmentation, Fog and Edge computing security, AI and Machine Learning threat detection, Zero Trust Architecture (ZTA), etc. as mentioned in [5]. This combination of IoT security technologies includes encryption, authentication, real-time threat detection, as well as ensuring secured communication between devices to ensure that with the growth of IoT ecosystems, networks become more secure through all these advanced technologies while battling the inherent vulnerabilities present in connected devices.

C. Objective and Scope

There are recent developments in embedded systems and operating systems focusing on improving performance, security, real-time processing, and energy efficiency, driven largely by the needs of IoT, automotive, and industrial automation sectors.

Embedded systems plays a crucial role int eh development of IoT ecosystems and consist of electronic devices with computational intelligence capabilities which are carefully designed to perform a single specific tasks. [2] These systems combines both hardware and software technologies together to improve performance and functionalities.

Embedded systems have developed its technologies in various sectors like Real-time Processing and Performance, AI and Machine Learning at the Edge, Low -Power and EnergyEfficient Designs, Enhanced Connectivity and Communication, Security in Embedded Hardware, etc. [6]. Embedded systems have wide variety of devices, such as Raspberry Pi, Arduino, FPGA devices, ODROID, ASUS Tinker Board, NanoPi M4, Potato AML CC, Banana Pi m64, LattePanda Alpha 864, BeagleBone Black, NVIDIA Jetson Nano, Pine64, ESP32, ESP8266 and much more. [2] These devices vary in their processing power, connectivity features, and application suitability, making them useful across different embedded system and IoT the development scenarios. Making it flexible and strengthening IOT environment. [7]

The Operating System (OS) employed in the ecosystem of IoT, initially, has its roots in Embedded and has undergone major developments like RTOS, Embedded Linux, Containerization and Virtualization, Power Efficiency in OS, and AI Integration in OS. It enables the systems to accomplish the specific requirements pertaining to the performance and power. [4] The role of the Operating System (OS) in Embedded IoT is pivotal as it grants the management of the hardware resources of IoT devices, allowing secure communication with the objects, and creating a stable environment for applications to run. Embedded IoT systems the OS must carry the duties of limitations of resources, must ensure real-time operation, and to maintain system reliability at the same time. [4]

IoT is the foundation of many critical sectors such as smart cities, industrial automation, healthcare, and agriculture and

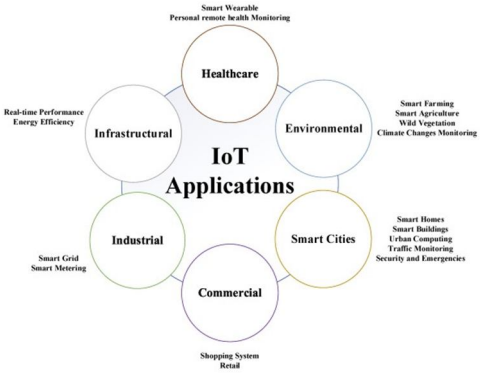


Fig. 2. IoT application sectors. [4]

is used to optimize processes, enhance efficiency, and drive innovation. The wide-scale deployment of IoT will make sure that a plethora of different locations always have access to realtime data and automation, but they must first and foremost do it securely to prevent unauthorized access, data breaches, and cyber threats. To make these environments globally available with security, IoT deployments should leverage cloud platforms such as AWS IoT, Azure IoT, etc., edge computing for local processing and implement strong and reliable security measures like encryption, secure boot, and device authentication to ensure data integrity and privacy across borders.

Canonical is the company behind Ubuntu Core, a lightweight OS designed for secure and scalable IoT deployments, with support for over-the-air updates and containerization via snaps. This OS is base on Ubuntu LTS (Long-Term Support). The objective is to deploy these IoT environments securely using vast variety of embedded devices for various scalable options. [2] This make the IoT environment more scalable which refers to ability to expand and manage large networks of connected devices efficiently as the system grows.

II. RELATED WORK

A. *Existing IoT running structures*

To correctly region Ubuntu Core inside the IoT landscape, it's very useful to take a look at it with several famous going for walks systems that can be developed in IoT structures.

1) *Raspbians:* Raspbian is a Debian-unique primarily based truely stable device designed in particular for the Raspberry Pi. It is specifically widely recognized among hobbyists and educational settings due to its individual-stunning interface and sturdy net software assets. However, it does not offer the strong competencies that the corporation's business enterprise customers might also choose, along with high levels of dependable protection and modern day strategies This limits its software to vital IoT regions of security and security is vital. [8]
2) *OpenWRT:* OpenWRT is mainly used for embedded gadgets in network routing and strolling gadgets just at the cutting edge of the day are best Linux-based. One of its key strengths is its premium customization certification, permitting customers to tailor the device to their unique desires. However, this flexibility calls for ordinary technical intensity, which can be an obstacle for the ones searching for plausible answers. [9] A study following the method of Morabito et al. (2015) spotlight the compatibility among flexibility and ease of use, suggesting that the latter, like Ubuntu Core, can be greater outstanding for customers who need to keep away from the complexities generally related to OpenWRT. [8]
3) *Windows IoT:* Microsoft's Windows IoT platform gives seamless integration with its unique functionality, making it a natural choice for firms already invested in Microsoft ecosystems, however this comfort comes with a cost. The platform struggles with hardware compatibility problems and lacks the open supply flexibility that many builders crave. This proprietary nature can stifle innovation and flexibility, especially in exclusive IoT environments where a couple of hardware sorts are required. [10]
4) *Ubuntu Core*

- The Snap Package System takes security to the next stage by means of retaining applications and services separate.This level of prevention protects the entire system from vulnerabilities, making Ubuntu Core an attractive choice for secure IoT deployments. [11]
- Ubuntu Core with atomic updates checks whether updates are complete or not, greatly reducing the risk of system instability due to incomplete updates This reliability is very important for IoT systems has maintained their integrity especially in critical applications.
- Snap systems enable systems to run in isolated environments, simplifying deployment and management. This method of loading not only greatly simplifies the integration of new applications but also simplifies the whole process. [12]

B. *Finding scalability in IoT operating systems*

Previous studies have delved into the flexibility of various IoT operating systems, including one such study that attempted to unlock the secrets of resource efficiency and user flexibility, conducted by Morabito et al around. [8], indicated that containerization provided high profit margins in these areas. These developments have far-reaching implications in IoT operating system development.

1) *Simplifying IoT device management:* Vogler et al. [9]¨ proposed a flexible framework for IoT device management, which emphasized the importance of efficient and secure switching mechanisms in large-scale deployments Their work is in line with Ubuntu Core design principles, in particular in terms of new connections and containerization.This interplay highlights the significance of an included method to IoT device management.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

2) *The manner forward: Progress and gaps:* Recent trends in IoT running systems have focused on improving safety, revolutionary strategies, and scalability. However, despite those advances, there are nevertheless gaps in requirements for deployment practices in heterogeneous IoT environments.

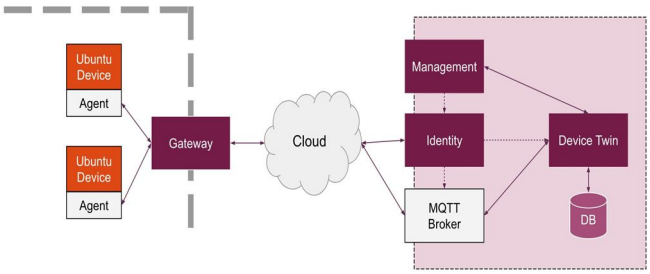


Fig. 3. Deployment of IoT service [13]

Ubuntu Core's new technique to containerization and tool management addresses a number of these demanding situations, however further studies is needed to evaluate its effectiveness in extraordinary actual-global situations As the IoT landscape keeps to evolve, it wishes to be pressured it is on this hole that the whole ability of IoT operating systems is unlocked.

III. ARCHITECTURE OF UBUNTU CORE

The architecture of Ubuntu Core is designed to be lean, basic, and secure. It is different from standard operating systems that manage applications, updates, and security. The architecture is based on 3 layers, which are the base layer, snap layer and snap layer. [2]

A. Base Layer

It provides only necessary components for the system to function. This allows Ubuntu Core to remain small, efficient, and secure. The kernel manages hardware interactions, device drivers, and basic system functionality.

B. Snap Layer

Snap is the heart of the Ubuntu Cores architecture; it is responsible for installing and updating applications and system components. Each snap includes everything that applications need to be run, like libraries. It is divided into different parts. [12]First is the core snap, which contains basic libraries and tools. Second is application snap. This is installed on device by user level. They can be anything from IoT applications to control systems, and the third is kernel snap; it contains a Linux kernel specific to the device and can be updated independently. [12]

C. Snapd Layer

The Snapd service manages installing, updating, and removing Snaps and handles automatic updates so that users should remain up-to-date with the latest features and security updates.

IV. DEPLOYMENT OF UBUNTU CORE IN IOT ECOSYSTEMS

A. Deployment Methodology

1) Deploying Ubuntu Core on IoT devices

- Prepare the IoT Device: To install Ubuntu Core on IoT ecosystems, the process consists of a group of the main actions ensuring secure equipment delivery and operation. Initially, the IoT device should be configured by deciding on a usable IoT hardware like Raspberry Pi, Intel NUC, ARM-based board, and others. Then, flashing the Ubuntu Core OS on the device's storage using tools like Balena Etcher. This will ensure that the OS is installed in secure, immutable format. [1]
- Initial Configuration: We have to boot the device to work properly. After the device boots, it needs to connect to a network for authentication and updates. Configuration may involve setting up Wi-Fi or Ethernet, depending on the deployment environment. Ubuntu Core relies on Ubuntu One for secure device authentication. Each device is registered under a user's Ubuntu One account, making it manageable from anywhere. [14]

- Installing and Managing Snaps: Snaps are self-contained, containerized software packages used to run applications on Ubuntu Core. They are central to Ubuntu Core's modular architecture, simplifying app installation and updates without impacting the core OS. [14] For example, Mosquitto can be used for MQTT communication, Node-RED for automation, or Docker for containerized applications on the IoT device.
- Over-the-Air (OTA) Updates: Because Ubuntu Core does transactional updates, if it fails to update completely, it will always roll back to the previous state without ever leaving the system unstable. This is important in IoT deployments where some of the devices are located in remote locations and cannot be readily accessed for a manual update. [14]
- Security Configurations: Ubuntu Core will implement secure boot-it will ensure that only trusted software is allowed to run on the device and will guarantee full disk encryption, ensuring data stored on the device cannot be misused. The supporting integration of the Trusted Platform Module (TPM) will continue for verifying device integrity and secure identity management. This will be an important feature in the industrial IoT; the trustworthiness of the device has to be assured. [1]
- Scaling with Cloud Integration: Ubuntu Core integrates with cloud platforms like Azure IoT, AWS IoT, and Google Cloud IoT for device management, monitoring, and data analytics. In this integration, thousands of devices can be connected to a central system without much hassle. [15] Canonical IoT management platform provides methods for remotely controlling fleets of devices, updating, and monitoring the health of devices in real time.
- Automation and Scaling: This can be scaled up with the help of DevOps tools like Jenkins or Azure DevOps, which can deliver the automatic deployment of updates to the IoT devices. Automation is indispensable for managing large IoT networks, minimizing manual intervention, and enabling consistency in the software of all devices.

Cloud Provider	Cloud Platform	Relevant IoT Services
Microsoft	Azure	Azure IoT Hub Azure IoT Edge Azure IoT Central
Amazon	AWS	AWS IoT Core AWS IoT Device Management AWS IoT Greengrass
Google	Google Cloud	Cloud IoT Cloud IoT Core Cloud IoT Edge

Fig. 4. Popular IoT cloud services and their providers [4]

2) *Integration with cloud platforms:* Integrating Ubuntu Core with cloud platforms like Azure IoT, AWS IoT, and Google Cloud IoT is critical for scaling, managing, and securing large deployments of IoT devices. These cloud platforms offer a range of services that simplify device management, data analytics, and remote monitoring, making it easier to oversee and control fleets of devices running Ubuntu Core. [15] The whole management and monitoring of IoT devices can be done with Azure IoT. Then Ubuntu Core can get plugged into the Azure IoT Hub to provide a very streamlined approach to device registration, secure communication, and data analytics. AWS IoT Core provides powerful tools for handling Ubuntu Core devices at scale. AWS IoT services make easy handling of communication, data management, and integration with services, such as Lambda, S3, and DynamoDB. [1] The Google Cloud IoT Core creates the possibility for a flexible and secure environment in which to manage high deployments of Ubuntu Core. Integration into BigQuery, Pub/Sub, and AI/ML tools will create ample benefits concerning advanced data analytics and machine learning for Ubuntu Core devices. Challenges in Integration

- Data Latency and Bandwidth Management: Although cloud integration enables scaling, it sends huge volumes of data from Ubuntu Core devices to the cloud, which may render them fairly latency-prone and bandwidth-consumptive. [1] This will be significant operational challenges to applications requiring in real time processing of data, including automated vehicles or industrial automation.
- Security and Privacy:
- Securing communication between Ubuntu Core devices and cloud platforms is critical. Each cloud provider implements different authentication and encryption mechanisms, so it's essential to configure security settings properly to prevent unauthorized access or data breaches.
- [15]
- Multi-Cloud Integration: Some IoT deployments may require integration with multiple cloud platforms (e.g., combining AWS and Google Cloud). Managing this complexity, especially with different APIs and data management practices, poses a challenge for developers and administrators.
- Device Provisioning at Scale: With the rapid increase in the number of IoT devices, providing provisioning to them securely and efficiently has become a great challenge. Cloud platforms like AWS and Azure provide device management tools, but managing configurations and updates for thousands of devices still poses a complex issue. [15]

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

3) *Configuration and provisioning of devices for largescale deployments:* Configuring and provisioning of devices for large-scale Ubuntu Core deployments requires creating a sequence of steps to ensure that each IoT device is securely set up, connected to the network, and ready for remote management. Devices are first flashed with the Ubuntu Core OS and authenticated with services such as Ubuntu One or one set of device-specific credentials like X.509 certificates or JSON Web Tokens (JWTs) meant for secure cloud integration, for example, with Azure IoT, AWS IoT or Google Cloud IoT. [16] The deployed devices, after registration, are provisioned by adjusting key parameters such as network settings, device roles, and application installations through snaps (containerized packages of software). Updates and configurations are automated by tooling such as DevOps pipelines, including Jenkins and Azure DevOps, to make provisioning batch-clicable for thousands of devices. The integrated cloud platforms provide devices that can now be monitored and updated remotely along with secure OTA management, helping scale up and deploy comprehensively, but with a constant attitude toward IoT environments, which may be different and distributed. [16]

B. Automation in Deployment

Automation for Deployment of Ubuntu Core in IoT Ecosystems is the most critical management tool, providing largescale networks of devices and minimizing manual intervention as well as inconsistent update delivery across devices. DevOps tools such as Jenkins, GitLab CI, or Azure DevOps will be able to automate the complete pipeline of deployment for Ubuntu Core devices, all the way from provisioning to OTA updates and configuration. All these tools help a developer conduct continuous integration/continuous deployment practices; they will automate the building of custom snap packages, test them, and proceed directly to deploy it to thousands of devices. Automation helps the tool ensure constant software versions, ensures devices are safe with timely updates, and minimizes human errors prevalent in manual deployment. It also increases scalability; changes and configurations can be rolled out in parallel across a number of devices regardless of location. Automation, for instance, ensures running and security within industrial IoT or smart city use cases without having to intervene physically by human hands- reducing operational cost and raising system uptime. [17]

V. PERFORMANCE EVALUATION

1) *Ecosystem:* Because of the open source operating system, the public can give valuable feedback for any bug or what changes will be required.
2) *System Resource Utilization:* Due to the lightweight OS, the Ubuntu core should give lower CPU utilization during underload.
3) *Boot Time:* Ubuntu Core is a thin and basic operating system, so the boot time required is very low compared with other OS.
4) *Security:* This Ubuntu core is especially made up for security of internet of things (IoT) devices. It is built on the concept of snaps. Snaps provide strong security. Related to security, one of the features of Ubuntu Core is automatic updates that improve security and reduce maintenance costs. [2]
5) *Power Consumption:* For embedded and IoT systems, low power consumption is an important thing. Ubuntu Core should have a smaller power footprint, and for battery-powered devices it can be used for longer operational lifespans. [11]
6) *Network Performance:* In operating systems, network performance is a key factor. Ubuntu cores have throughput and reaction time over different network interfaces like wifi and Ethernet. It also uses communication protocols like MQTT, HTTP, and CoAP.
7) *Reliability and Stability:* Hold stability for a long time. It can be difficult, especially for IoT devices. It is expected to run continuously for a long time due to regular updates. Ubuntu core can handle stability for a long time.

VI. SECURITY CONCERNS IN BIG-SCALE APPLICATIONS:

As our international connections become more and more associated, the amount of IoT devices is developing suddenly, supplying a huge sort of ability protection threats. [2]

A. Equipment robbery: Uninvited tourist

Imagine coming home to find out your smart thermostat has been blown out. A stranger changes your temperature settings, or worse they could get entry to your private statistics. Without robust safety features, even the maximum delicate IoT gadgets can turn out to be objectives for cybercriminals. It's like having an uninvited visitor in your house, controlling your gadgets and getting access to your personal facts. [18]

B. Network Attacks

Man-in-the-Middle (MitM) Attacks: Picture this: a hacker intercepts the communication among your clever fridge and the cloud, changing the data being sent from side to side. This could cause fake signals or even compromise your food safety. [18] Denial of Service (DoS): Now believe being unable to get right of entry to your property safety device because a flood of malicious visitors is overwhelming it. As Vogler et al. (2015)¨ factor out, sturdy community designs are important to defend towards such disruptions. [18] Unauthorized Access: Leaving

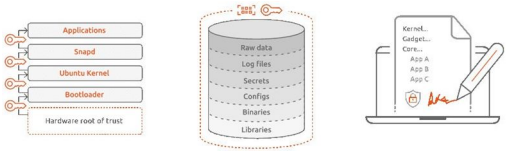


Fig. 5. Ubuntu Core secure boot flow diagram

vulnerable passwords on devices is like leaving your front door unlocked. Unauthorized customers can waltz in and get entry to touchy facts. Research suggests that enforcing sturdy identity management is critical for keeping undesirable site visitors out. [18]

C. Security Features of Ubuntu Core: Guardian of IoT

- TPM (Trusted Platform Module): Think of TPM as a middleware vault installed on your device. It purchases very essential gadgets and ensures that essential information continues to be beneath wraps. Relying on hardware protection offers a more acceptable function for our security.
- Secure Boot: This feature works like a trusted bouncer, allowing you to launch quality hosted software application software utilities even when your device is plugged in. If an attacker tries to install a malicious software application, secure boot you Properly outside will be protected at the doors , make sure your equipment goes in a sturdy bag completely inexperienced App isolation and snaps: Ubuntu Core uses ”snaps”, which are also like small programs for separate apps. If one app is shut down, the others live safe. This isolation is vital in a worldwide environment in which protection breaches can permeate the complete network.
- Automatic updates: Forgetting to change your gadgets can be wishful thinking, however, with automatic updates, you don’t want to be scared! Your devices will assemble the latest safety bands without trying to bulk up your wrist. This approach allows you to maintain a tight grip on your digital presence.

VII. CASE STUDY OR PRACTICAL EXAMPLE

The industrial technology and IoT solutions leader Bosch has chosen Ubuntu Core for the power of smart appliances across industries. These could be smart home appliances, manufacturing, or automotive systems. Bosch offers a massive number of portfolios in terms of the IoT devices up for usage. This, however, necessitated a solid and scalable platform that can support a wide range of hardware configurations. The Ubuntu Core was meant to fill that gap. Its lightweight, containerized design used by Snaps satisfied this requirement. Bosch was able to reduce the complexities of deploying IoT devices and got an almost uniform functionality over many varied environments. [19]

One of the big issues that Bosch was facing was that it could not update thousands of devices, whose hardware specifications differed greatly in size, ranging from tiny sensors to large industrial machines. Bosch leveraged Ubuntu Core’s cloud-based management and OTA update for a seamless, centralized control of its fleet of devices and updates with minimal downtime. This put Bosch in an early position to increase its operations without needing to concern itself with the complexity of rolling up upgrades across that diverse device mix. [19]

Another area of high importance for Bosch is security, particularly in the areas of manufacturing and automotive industries, since hacks on devices pose the potential risk of losing a lot of money and debilitating operations. The risks that Ubuntu Core was facing were hence mitigated by its sandboxed Snaps and transactional updates. In case updates fail, the automatically rolling back ensures no occurrence of downtime or compromised devices. [19]

Another benefit that Bosch was able to cash in on was Ubuntu Core’s support for edge computing. With applications running natively and directly on devices located at the edge, Bosch could decrease the latency while optimizing bandwidth usage and enhance the processing of data in real-time. This was a very critical application to take into account sectors like automotive and industrial automation, which are often timesensitive, requiring local processing of data in order to reach prompt decisions. [19]

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

Therefore, in summary, utilization of Ubuntu Core by Bosch in its deployments has given the company considerable enhancement concerning issues related to scalability, security, and operational efficiency of its deployment. Since devices could be remotely and securely managed and updated, Bosch was able to maintain high standards of performance and reliability throughout its IoT ecosystem, meaning that its devices were always updated with the requirements necessary while at the same time ensuring that the security conditions were maintained. Through this relationship, Bosch has been able to continue leading in the IoT space while being able to offer innovative solutions to a myriad of industries across its client base. [19]

A. About Canonical

Canonical is a UK-based software company and would perhaps best be explained by their flagship product, Ubuntu; it's one of the biggest distributions in use worldwide. Mark Shuttleworth founded Canonical in 2004 with a goal to create an open source solution for cloud, IoT and enterprise computing. Ubuntu itself is one of the most widely deployed distributions in many sectors mainly because of its flexibility, security, and scalability. Canonical, too advances and maintains Ubuntu Core, a lightweight version of Ubuntu for the Internet of Things devices. Its list of features included containerized applications, secure updates, and device management. [19]

B. About Rexroth

Bosch Rexroth is part of the Bosch Group, world's leading supplier of drive and control technologies. Rexroth specializes in advanced automation and industrial solutions ranging from hydraulics and electric drives to mobile systems and machinery automation. A firm with great focus on Industry 4.0 and IoT, Rexroth helps industries modernize their current production environment with smart technologies that will make them efficient, reduce downtime, and make them more flexible as they run through the manufacturing processes. With integrated technologies such as Ubuntu Core, Rexroth allows scalable and secure IoT solutions, particularly in industrial automation. [19]

VIII. DISCUSSION AND FUTURE DIRECTIONS

A. Discussion

Such demand comes from the need for a light, secure, and flexible OS for the connected devices. The primary advantage of Ubuntu Core is its reliance on snaps, which is a modular, containerized package management system. The snap system's architecture is what protects the applications and services from one another and, thus, system security is redesigned. Likewise, the transactional updates in Ubuntu Core allow support for over-the-air updates with no system downtime and minimal chances of the system becoming corrupted. It was these aspects that guaranteed that Ubuntu Core would be a good fit for IoT applications where the reliability and security of the device were of utmost importance. [2] Ubuntu Core is one of the many precautions that cities are taking to guarantee safety in the growing smart city areas. This includes Secure Boot, Full Disk Encryption, and the use of a Trusted Platform Module (TPM) for the integrity of devices. These aids IoTs are protected from threats such as access by nongenuine entities or hijacking of the devices. With every critical infrastructure like smart cities, healthcare, and industrial automation starting popping up with IoT devices everywhere, the need for scalable, secure solutions has never been greater than now. [18] However, Ubuntu Core deployment brings some problems with itself. Thus, the most important of them is the hardware compatibility issue. Ubuntu Core indeed supports several platforms like the Raspberry Pi and Intel NUC, but perhaps not as strict or individually designed IoT devices, and it becomes much harder to keep compatibility with most of the devices. This may lead to more time and effort spent on the image making process or driver integration. [16] One of the concerns about IoT networks is network stability. In many cases, the deployment of IoT is being made in places where it cannot be guaranteed that the internet connection will be strong. Rural areas, industrial locations, or even remote monitoring stations suffer from such issues. This in turn makes the function of updates and the administration of the machines more difficult. Edge computing and offline updates are the solutions to such issues but at the same time, they are adding complexity and requiring additional infrastructure. [17] Least expensive scalability is the most important part.
Ubuntu Core is an undoubtedly free open-source operation system but managing immense data storage processes in the cloud, especially on storage, data transfer, and processing, can be an expensive thing to do. As the number of devices grows rapidly, you will be required to develop plans to reduce the cost through edge computing or pay-as-you-go cloud services.

B. Future Directions

Looking forward, the role of Ubuntu Core in IoT ecosystems is about to undergo dramatic change, driven by the timerelated advances coming in communication, security, and computational technologies.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

- 5G and Edge Computing Integration: The emergence of 5G networks is poised to transform how IoT is deployed with ultra-low latency, higher bandwidth, and unprecedented numbers of devices that can be serviced. This greatly enhances the scalability of IoT environments particularly, which require real-time data processing and decision-making, such as in cases of autonomous vehicles, smart grid systems, and remote healthcare. Additionally, Ubuntu Core, with 5G, offloads some of the processing to the edge devices so that dependence on the cloud infrastructure is relieved, and real-time responsiveness is maintained. This, in turn, also enables local data processing on Ubuntu Core devices, lowering latency and improving efficiency, especially in bandwidth-constrained or latency-sensitive applications.
- AI at the Edge: Another promising direction for Ubuntu Core is AI and machine learning at the edge-the integration of AI and machine learning capabilities directly into the edge Internet of Things devices. Now, with frameworks such as TensorFlow Lite and TinyML, the same can be enabled to run real-time inference on Ubuntu Core devices, making the device much smarter and enabling that level of decision-making or to act on a decision without depending on cloud-based processing. It becomes very helpful for use cases like predictive maintenance, where machines can self-diagnose based on historical data, and can even act ahead in time by sending alerts about potential failures. The scaling will improve with reduced data transmission requirements since running AI models locally is possible. [17]
- Zero Trust Security Models: Traditional security models, which assume that devices inside a network can be trusted, will be insufficient as the IoT ecosystem grows in complexity. Zero Trust Architecture (ZTA) will play an important role in IoT deployments. Under the Zero Trust model, every device, user, and network interaction is authenticated, authorized, and continuously validated, regardless of whether they are inside or outside the network perimeter. [18] The existing inherent security features available through Ubuntu Core—such as secure boot, encrypted file systems, and snap confinement —provide a foundation for realizing Zero Trust models. To ensure that each IoT device running Ubuntu Core has continuous verification, the threat of a breach can be greatly mitigated within missioncritical environments such as healthcare or industrial automation.
- Standardisation and Interoperability: For full realization in large-scale IoT deployments, standardization and interoperability are going to significantly surface. As of now constituted, the IoT ecosystem is fragmented with different devices using different platforms and communicating through various protocols. If Ubuntu Core is going to deploy in diverse environments, it will have to support and subscribe to open standards that assure interoperability across a wide range of devices and ecosystems. Other open standards will include all forms of communication protocols such as MQTT, CoAP, or LwM2M, ensuring free communications between devices, standardized APIs will make the creation of applications that can easily work together interoperable with the cloud platform and others, avoiding vendor lock-in.
- Long-Term Sustainability and Green IoT: Another important direction in the future for IoT would be sustainability. With the more addition of devices to IoT, it would be leaving a wider environmental footprint. In this regard, Ubuntu Core would be able to contribute to the concepts under Green IoT, allowing efficient optimization of energy in devices and shrewd resource management. By leveraging the idea of edge computing, Ubuntu Core may significantly conserve much power on continuous cloud connectivity, thus saving significant amounts in energy through data transmission and processing. Further, this can improve sleep modes as well as dynamic power management, thus reducing the power consumption of the devices that comprise the IoT, especially on battery-powered sensors and remote devices.
- Decentralized IoT Networks (Blockchain): Another avenue where Ubuntu Core is destined for IoT ecosystems includes the utilization of blockchain as well as other decentralized technology to provide a decentralized structure-an absence of single points of failure-measured against the mission-critical nature of such applications as supply chain tracking, smart grids, and critical infrastructure monitoring. Blockchain can be used for device authentication management, secured data exchanges, and smart contracts in smart automated transactions between IoT devices. Ubuntu Core may include blockchain-based frameworks, making IoT networks immune to attacks and fraud, hence trustless. [18]

IX. CONCLUSION

Ubuntu Core deployment in IoT ecosystems is very promising and scalable in managing connected devices securely and efficiently. In our exploration, we engaged with key aspects in the deployment process-from preparing the hardware and installing the OS to integrating Ubuntu Core with cloud platforms such as Azure IoT, AWS IoT, and Google Cloud IoT. The Ubuntu Core snap-based modular architecture, along with the transactional OTA update capabilities and strong security features like secure boot, TPM integration, and full disk encryption, offers a proper strong IoT framework, especially in applications or environments where high reliability and minimal downtime are required.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

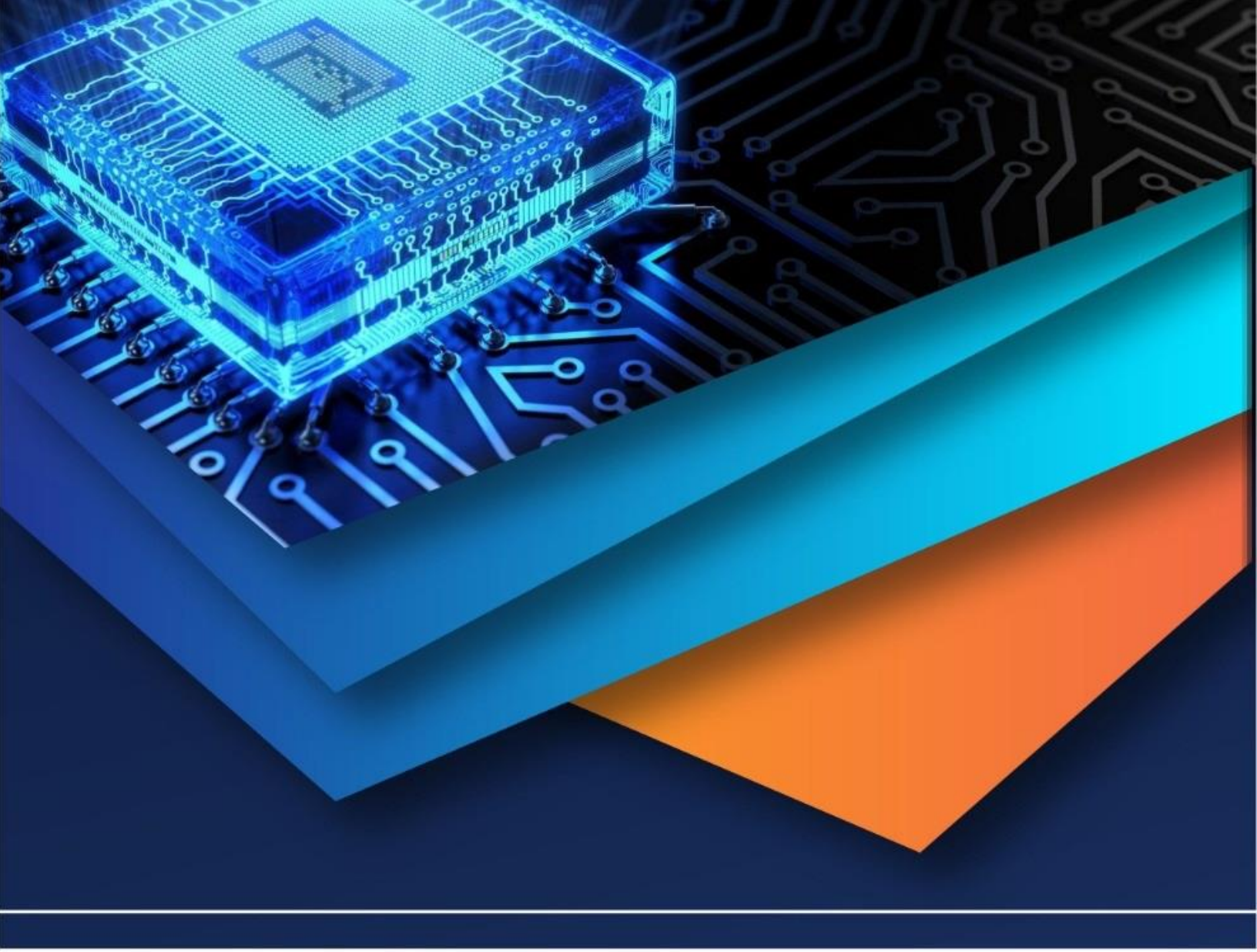
Although the system offers plenty of benefits in terms of security and ease of management, scale deployments bring about some challenges that need to be addressed. These include hardware compatibility issues, the instability of a network, and lack of efficiency in cost management upon scaleup. Meanwhile, edge computing technologies in association with automation by adopting CI/CD pipelines and real-time monitoring technologies are some of the effective methods to enhance the scalability of Ubuntu Core in diverse IoT scenarios.

For the future, upgrading to 5G, IoT, AI at the edge, Zero Trust security models, and decentralized technologies such as blockchain, will be the core elements in overcoming present limitations and broadening applicability scope of Ubuntu Core. Therefore, the overall capability will provide increased security in real time, combined with better control over the device, especially in applications for smart cities, industrial automation, healthcare, and critical infrastructure.

In summary, Ubuntu Core has proven to be one of the best players in the IoT ecosystem. It is secure, scalable, and fits perfectly with emerging technologies, thus turning out to be a critical component for the future of IoT deployments in combining cloud and edge solutions along with its security-first approach. With the continuous evolution of the IoT landscape, Ubuntu Core will be an essential tool in realizing and growing intelligent, interconnected systems. It is adaptable, scalable, and highly secure.

REFERENCES

[1] S. Trilles, A. Gonzalez-P´ erez, and J. Huerta, “An iot platform based´ on microservices and serverless paradigms for smart farming purposes,” Sensors, vol. 20, no. 8, p. 2418, 2020.
[2] A. D. Echeverria, M. A. Pinilla, and H. R. C. Mora, “Securing the iot: An in-depth analysis of ubuntu core hardening measures using cis lts guide,” in Interdisciplinary Conference on Electrics and Computer, INTCEC 2024. Institute of Electrical and Electronics Engineers Inc., 2024.
[3] S. Al-Sarawi, M. Anbar, K. Alieyan, and M. Alzubaidi, “Internet of things (iot) communication protocols,” in 2017 8th International conference on information technology (ICIT). IEEE, 2017, pp. 685–690.
[4] T. Blafield, D. H¨ astbacka, and J. Hankkila, “Automation of deployment¨ process for iot edge devices creating deployment pipeline for producing generic iot edge device os images utilizing tpm attestation,” Master’s thesis, 2022.
[5] K. M. Sadique, R. Rahmani, and P. Johannesson, “Towards security on internet of things: applications and challenges in technology,” Procedia Computer Science, vol. 141, pp. 199–206, 2018.
[6] S. C. Meka, S. Achan, and R. G. Pettit, “Real-time embedded monitoring technologies in modern healthcare systems: A survey,” in 2024 IEEE 27th International Symposium on Real-Time Distributed Computing (ISORC). IEEE, 2024, pp. 1–6.
[7] C. Hardware, “Embedded systems,” in 14th International Workshop, 2010.
[8] R. Morabito, J. Kjallman, and M. Komu, “Hypervisors vs. lightweight¨ virtualization: A performance comparison,” in 2015 IEEE International Conference on Cloud Engineering (IC2E). Tempe, AZ, USA: IEEE, 2015, pp. 386–393.
[9] M. Vogler, J. M. Schleicher, C. Inzinger, and S. Dustdar, “A scalable¨ framework for provisioning large-scale iot deployments,” ACM Transactions on Internet Technology (TOIT), vol. 16, no. 2, pp. 1–20, 2015.
[10] D. Borycki, Programming for the Internet of Things: Using Windows 10 IoT Core and Azure IoT Suite. Microsoft Press, 2017.
[11] D. Carson, “The ubuntu operating system,” Shepherd University, Department of Computer Sciences, Mathematics, and Engineering, P.O. Box 3210, Shepherdstown, WV 25443, Tech. Rep., 2022, contact: Dcarso03@rams.shepherd.edu.
[12] A. B. Dhule and D. Y. Chirayil, “Sustainably nurturing a plant (snap) using internet of things,” in Computational Intelligence: Select Proceedings of InCITe 2022. Springer, 2023, pp. 765–775.
[13] C. Ltd., “Ubuntu iot deployments - github repository,” 2023, accessed: October 5, 2024. [Online]. Available: https://github.com/canonical/iotdeploy?tab=readme-ov-file
[14] N. Ferry and P. H. Nguyen, “Towards model-based continuous deployment of secure iot systems,” in 2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C). IEEE, 2019, pp. 613–618.
[15] C. Stergiou, K. E. Psannis, B.-G. Kim, and B. Gupta, “Secure integration of iot and cloud computing,” Future Generation Computer Systems, vol. 78, pp. 964–975, 2018.
[16] M. A. Lopez-Pe´ na, J. D˜ ´ıaz, J. E. Perez, and H. Humanes, “Devops´ for iot systems: Fast and continuous monitoring feedback of system availability,” IEEE Internet of Things Journal, vol. 7, no. 10, pp. 10695– 10707, 2020.
[17] F. Li, M. Vogler, M. Claeßens, and S. Dustdar, “Towards automated¨ iot application deployment by a cloud-based approach,” in 2013 IEEE 6th international conference on service-oriented computing and applications. IEEE, 2013, pp. 61–68.
[18] I. Rule4, “Ubuntu core cybersecurity analysis,” Rule4, Inc., 3002 Bluff Street, Suite 100, Boulder, CO 80301, Tech. Rep., 2020, white paper. [Online]. Available: https://www.rule4.com
[19] C. Ltd., “Bosch rexroth adopts ubuntu core and snaps for app-based ctrlx automation platform,” https://canonical.com/blog/bosch-rexroth-adoptsubuntu-core-and-snaps-for-app-based-ctrlx-automation-platform, 2020, accessed: October 5, 2024.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)