*IOSR Journal of Computer Engineering (IOSR-JCE)*
*e-ISSN: 2278-0661, p-ISSN: 2278-8727, Volume 26, Issue 6, Ser. 2 (Nov. – Dec. 2024), PP 18-25*
*www.iosrjournals.org*

# Appization™ @ Neurahub™: Leveraging The App Store Model For AI Functions On Edge Devices_ Case Study Of Indoai AI Camera

Vivek Gujar,
*Founder-Director, Indoai Technologies P Ltd, Pune, India*

Ashwani Rathore,
*Founder-CEO, Indoai Technologies P Ltd, Pune, India*

***Abstract***
*The " Appization™@ NeuraHub™ " framework applies the successful AI app store model for mobile handsets to AI functions on edge devices, aiming to create an adaptable, scalable and customizable platform for deploying, managing, and updating AI models on specialized devices like IndoAI's Edge Cameras. This paper provides a comprehensive examination of the Appization™@ NeuraHub™platform, proposing it as an ecosystem where AI models can be installed, managed, and updated independently. NeuraHub is a comprehensive ecosystem where AI models can be deployed, managed and updated independently for developers. The framework introduces standardized protocols, a developer-friendly marketplace and a microservices architecture that streamlines AI model deployment on edge devices, overcoming current challenges and scaling edge AI for specialized applications.*
***Keywords:*** *Appization, NeuraHub, Edge Intelligence, AI Camera, App Store, AI Model, IndoAI*

---

Date of Submission: 26-11-2024 Date of Acceptance: 06-12-2024

---

## I. Introduction

In mobile technology, the app store model has revolutionized how applications are distributed, allowing seamless updates, on-demand downloads and customization on a standardized platform. Millions of apps are on the market[1], actively used by more than 300,000 companies in the United States. Globally, the app economy has been reported to be a $1.7 trillion ecosystem. Over the past 10 years, the usage of smartphones skyrocketed from 27 percent of persons ages 3 and older to 70 percent.

Traditionally[2] all data are directed to a central hub for processing, which is inadequate for businesses overwhelmed with data since it places excessive strain on the network. Here, edge computing has quickly emerged as a fundamental component of modern infrastructure for the AI era. By providing inferencing capabilities at the edge, operators can deliver fast, personalized and context-aware solutions to customers. This strategy not only enhances user experiences but also opens up new revenue opportunities, driving growth and competitiveness in the telecommunications sector.

According to Basole et al [3] the mobile service ecosystem is in a state of rapid and fundamental transformation and the emergence of smart-phones, massive improvements in mobile network infrastructure, significant advances in payment platforms and an explosive demand for mobile applications have created enormous revenue opportunities for all players in the ecosystem. According to Wang et al [13]Mobile Edge Computing pushes compute intensive assignments to the edge of the network and brings resources as close to the endpoints as possible, addressing the shortcomings of mobile devices with regard to storage space, resource optimisation, computational performance and efficiency.

According to Redhat[4],Mobile edge computing is a type of network architecture that provides cloud computing capabilities and an IT service environment at the edge of the network. While NVIDIA defines, Edge AI is the deployment of AI applications in devices throughout the physical world. It's called "edge AI" because the AI computation is done near the user.

The power of running AI application on the edge device is reshaping the future landscape for surveillance cameras and vision applications [5]. Carlos Cob-Parro et al [6] proposed system of of computing on the edge of nodes the intelligence of the system(smart video surveillance) is distributed in multiple nodes, where each one can include a camera and a processing system that performs simple tasks before sending the information to the operator, facilitating its work. AI-powered edge computing[7], where artificial intelligence algorithms are deployed directly on edge devices, enabling them to analyze data, make decisions, and act autonomously.

Edge devices, particularly AI-powered cameras like IndoAI's, face a similar need for modular AI functionalities in fields as varied as security, industrial monitoring and data analytics. Currently, such devices often lack modularity, requiring either pre-installed AI models or significant technical expertise for updates, which restricts scalability and hinders adaptability. To address these challenges, this article introduces "Appization™ @ NeuraHub™ " a framework designed to provide edge devices with a flexible, app store-inspired ecosystem that allows for plug-and-play AI model deployment.

## II. Framework Of Appization™ @Neurahub™ Model For Edge Devices

Edge computing, according to Mohan Harish et al [15] requires app models that are both efficient and conscious of resource usage. Dynamic resource allocation appears to be a viable strategy for improving energy efficiency in edge computing settings. The authors, Oumayma Jouini et al [8] state that it is advisable to spread data processing across IoT devices, gateways, fog and cloud computing, they add, an IoT device can carry out initial data processing with lightweight embedded models and transfer only the essential data to the cloud for further analysis. This idea emphasizes the implementation of ML models throughout all tiers of the network, utilizing the hierarchical organization of the IoT framework. In their work, the authors, Ran et al [14] design a framework that ties together front-end devices with more powerful backend "helpers" (e.g., home servers) to allow deep learning to be executed locally or remotely in the cloud/edge.

Thus**, An effective framework should emphasize:**
Modularity: Decompose applications into distinct, self-sufficient modules for streamlined updates and enhancements.
Resource Efficiency: Develop applications that reduce memory and processing power demands, ensuring seamless performance on limited devices.
Offline Capability: Allow applications to operate independently, without a continuous connection to the cloud.
Security: Enforce robust security protocols to safeguard sensitive information and thwart unauthorized access.
Real-time Processing: Tune applications for minimal latency response times, which is crucial for applications needing immediate action.
Scalability: Structure models to accommodate fluctuating workloads and adjust to varying device capabilities**.**
Interoperability: Guarantee compatibility across a range of hardware and software systems.

AI Integration: Embed machine learning models for smarter decision-making and predictive insights [8].

**Table 1.** A Comparison of Frameworks and Libraries for Edge Intelligence.

| Framework Library | Development Language | Edge Device Requirements | Open Source | Task | Applications |
|---|---|---|---|---|---|
| **TensorFlow Lite** | C++, Java | Mobile Embedded Device | Yes | Inference | Computer Vision [54], Object Detection [55]. |
| **Caffe2** | C++ | Multiple Platforms | Yes | Training and Inference | Image Analysis [56], Video Analysis [57]. |
| **Core ML** | Python | Apple Devices | No | Inference | Image analysis [58] |
| **MXNet** | Python, C++ | Multiple Platforms | Yes | Training and Inference | Image Recognition [59] Text Recognition [60] |
| **PyTorch** | Python | Multiple Platforms | Yes | Training and Inference | Image Recognition [61] Text Recognition [62] |
| **AWS IoT Greengrass** | Python, Node.JS, Java, C and C++ | Multiple Platforms | Yes | Inference | Precision Agriculture [63], Autonomous [64] Vehicles [65] |
| **Edge2Train** | C++ | MCUs supported by Arduino IDE | Yes | Training and Inference | Video Analysis [66] |
| **OpenEI** | – | Multiple Platforms | Yes | Training and Inference | Various Applications |
| **TensorRT** | C++ | NVIDIA GPU | No | Inference | Image Classification [67] |
| **DeepThings** | C/C++ | Multiple Platforms | Yes | Training and Inference | Object Detection |

**Framework Libraries for Edge Intelligence**

| Framework/Platform | Description | Key Features | AI Model |
|---|---|---|---|
| **TensorFlow Lite** | A lightweight version of TensorFlow optimized for mobile and embedded devices. | Enables on-device machine learning with low latency and minimal memory footprint. | TensorFlow models optimized for edge use<br><br>AI models are converted into TensorFlow Lite format using a model converter for optimized inference on edge devices. |
| **PyTorch Mobile** | A mobile version of PyTorch, a popular deep learning framework. | Allows for the deployment of PyTorch models on mobile and embedded devices. | PyTorch models for mobile deployment |

| | | | |
|---|---|---|---|
| | | | PyTorch models are exported and optimized for mobile deployment using TorchScript or Mobile Optimizer. |
| **Edge Impulse** | A cloud-based platform for developing and deploying machine learning models on edge devices. | Offers a user-friendly interface and a variety of tools for data acquisition, model training, and deployment. | Custom-trained edge-specific models<br><br>Models are created or uploaded, trained in the cloud, and deployed as lightweight edge-ready binaries to devices. |
| **TinyML** | A community-driven project focused on machine learning on microcontrollers. | Provides tools and libraries for training and deploying small, efficient machine learning models. | Microcontroller-optimized ML models<br><br>AI models are quantized and compiled into microcontroller-friendly formats using tools like TensorFlow Lite Micro. |
| **TVM (Tensor Virtual Machine)** | A deep learning compiler stack that optimizes models for various hardware platforms. | Enables efficient deployment of machine learning models on diverse edge devices. | Framework-independent compiled models<br><br>Models are compiled into highly optimized binaries using TVM's deep learning compiler for specific hardware. |
| **ONNX Runtime** | A high-performance inference engine that supports a variety of machine learning frameworks. | Can be used to deploy models on edge devices with minimal overhead. | ONNX-compatible AI models<br><br>AI models from various frameworks are exported into ONNX format and optimized for deployment using the runtime engine. |
| **Apache MXNet** | A flexible and scalable deep learning framework that supports a wide range of hardware platforms. | Can be used for both training and inference on edge devices. | MXNet-trained neural networks<br><br>Models are trained and exported in formats compatible with both edge devices and scalable hardware. |
| **AWS Greengrass** | An edge computing platform for managing device fleets and deploying applications remotely, streamlining IoT network operations. | Supports device management using MQTT or other messaging protocols, with optimized data transfer to the cloud. | Pre-trained IoT and edge AI model<br><br>AI Models are trained and exported in formats compatible with both edge devices and scalable hardware. |
| Compiled | | | |

The IndoAI's Appization™ @NeuraHub™ model brings the app store concept to the edge AI domain, adapting elements to cater to the unique requirements of real-time AI processing on edge devices.

**Structural Framework**

According to Wikipedia an app store, also called an app marketplace or app catalog, is a type of digital distribution platform for computer software called applications, often in a mobile context. While [9]defines App store as an online curated marketplace that allows developers to sell and distribute their products to actors within one or more multi- sided software. An API gateway[10] is an API management tool that acts as an intermediary between an API client and backend services. The role of API gateways[11] in software development, abstracting the complexity of microservices and simplifying the client's interaction with the application. A developer portal[12] provides developers with essential documentation, tools, resources and support for understanding and integrating software.

The IndoAI's Appization™ @ NeuraHub™ model consists of several components:

- AI Model Marketplace: A repository for AI models, allowing users to browse and install models based on their needs.
- API Gateway: Serves as the control center, managing requests between the marketplace, the device and core services.
- Core Services: Encompasses essential functions like authentication, authorization and user management.
- Developer Portal (NeuraHub): Enables third-party AI model developers to upload and manage models, fostering a vibrant developer ecosystem.

**Table 1: Key Components of the Appization™ @ NeuraHub™ Model**

| Component | Description |
|---|---|
| **AI Model Marketplace** | Central repository for selecting, downloading, and updating AI models |
| **API Gateway** | Manages communications, load balancing, and security |
| **Core Services** | Authentication, authorization, and model management |
| **Developer Portal** | Platform for third-party model development and contribution |
| **Edge Device (IndoAI Camera)** | Platform where models are deployed and processed |

**Architectural Flow and Model Deployment**

Authors Rongxu et al [16] have presented a method to secure microservices based on tokens using a REST API gateway in the edge computing environment as a microservice to provide a lightweight and secure computing framework at the edge of the network. The architecture is designed to facilitate dynamic AI model installation.

In the IndoAI case, Edge devices communicate with the core Appization™ @ NeuraHub™ infrastructure through the API Gateway. When a user selects a model from the AI Model Marketplace, the API Gateway manages the download, verifies permissions and deploys the model directly to the device.

**FLow 1: Appization™ @ NeuraHub™ Architecture**

[AI Model Marketplace 🛒] ←→ [API Gateway ⚙️]←→ [Core Services 🛠️] ←→ [Edge Device ර=] [Developer Portal 🌀] → [Container Registry] ←→ [API Gateway ⚙️]

Each model in the marketplace is stored as a containerized service, ensuring that it operates independently of others. This setup allows the platform to handle multiple AI models concurrently and facilitate updates without impacting device operation.

## III. Key Components And Functionality Of The Appization™ @ Neurahub™ Model

**AI Model Marketplace**

The marketplace is a user-friendly platform where various AI models can be explored, including facial recognition, fire detection, vehicle detection and other specialized models. Similar to mobile app categories (e.g., productivity, entertainment), Appization™ @ NeuraHub™ offers categories for AI models to simplify model selection. Each model can have metrics, ratings and feedback that guide user selection.

- Example Model Categories:
  - Security: Facial Recognition, Object Detection, Vehicle Recognition, Group Riot detection, Gesture recognition, etc
  - Industrial Safety: Fire Detection, Hazardous Object Detection, No Fall areas, No Go areas, etc
  - Analytics: Heat Maps, Activity Monitoring, People Counting, etc

**Modular Microservices-Based Architecture**

A Microservice [17] is a small or even micro independent process that communicates, acts and returns via messages through lightweight mechanisms.

Each model in Appization™ @ NeuraHub™ is designed as an independent microservice, which facilitates deployment, management and scaling. A microservices approach enables each AI model to function independently, eliminating the need for inter-model dependencies.

- Benefits:
  - Easy Scaling: Each model operates as a standalone service, allowing selective scaling.
  - Independent Updates: Models can be updated independently, reducing downtime.
  - Enhanced Security: Microservices are isolated, minimizing the potential impact of security breaches.

**Figure 2: Microservices Communication within Appization™ @ NeuraHub™**

[Core Services] ←→ [Facial Recognition Model] ←→ [API Gateway] ←→ [Edge Device]
[Core Services] ←→ [Fire Detection Model] ←→ [API Gateway] ←→ [Edge Device]

**API Gateway**

The API Gateway facilitates seamless communication across the platform. It routes requests, authenticates users and handles load balancing, ensuring a stable, secure, and efficient operation. It also acts as the point of access for developers, who can submit new AI models and updates to the marketplace.

**Developer Portal (**NeuraHub™)

Appization™ @ NeuraHub™'s developer portal is designed to foster a collaborative ecosystem where third-party developers can contribute AI models, similar to mobile app store development. It provides resources like documentation, SDKs, testing environments and a revenue-sharing model.

- SDKs and Documentation: Provides essential resources for AI model development.
- Revenue-Sharing: Incentivizes developers by offering a share of revenue based on usage.
- Testing Environments: Allows developers to test models in simulated environments before deployment.

**Table 2: Developer Ecosystem Comparison Mobile-app Vs Appization™ @ NeuraHub™**

| Feature | Mobile App Store | Appization™ @ NeuraHub™ Developer Portal |
|---|---|---|
| **Developer Support** | Comprehensive SDKs, Testing | AI-specific SDKs, Edge Model Testing |
| **Revenue Generation** | Revenue sharing on app purchases | Usage-based revenue share for model usage |
| **Community Engagement** | Forums, workshop and events | AI-focused webinars, developer community |

## IV. Workflow And Process

**Model Deployment Process**

According to Microsoft [22], Deploying a model means taking a trained ML model, packaging it (like as a container image or a pipeline), and setting it up for inference. One of the main findings of the study by Sonntag et al [21] is that the model developed is the first to calculate an AI deployment capability and fully weight the indicators against each other and developed following ( see below image) AI deployment Model called SMMT model:

| Criteria \ Maturity models | Algorithmic business maturity model (Gentsch, 2019) | AI maturity map (Kreutzer & Sirrenberg, 2019) | AI maturity model (Xavier Health, 2020) | AI readiness check (TMForum, 2020) | AI maturity model (Alsheibani et al., 2019) | Auditing artificial intelligence maturity model (Fukas et al., 2021) | AI algorithm maturity (Niewiadomski et al., 2019) | Logistics 4.0 maturity model (Oleśków-Szłapka et al., 2019) | Industry 4.0 maturity (Santos & Martinho, 2020) | AI maturity model (Schuster et al., 2021) | Sustainability maturity model (Vásquez et al., 2021) | SMMT maturity model |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Number of maturity levels | 4 | 2 | 5 | 5 | 5 | 5 | 5 | 5 | 6 | 5 | 4 | 5 |
| Number of maturity dimensions | 5 | 8 | 4 | 6 | 4 | 8 | 4 | 3 | 5 | 7 | 3 | 5 |
| Number of maturity indicators | 33 | 8 | 15 | 54 | 0 | 40 | 25 | 25 | 35 | 0 | 42 | 29 |
| Theory model already applied in practice, tested, and evaluated | ○ | ○ | ● | ● | ○ | ○ | ● | ○ | ● | ○ | ○ | ● |
| Is the maturity model methodically derived | ● | ● | ◐ | ◐ | ● | ● | ● | ● | ● | ● | ● | ● |
| Includes a building step model | ◐ | ◐ | ● | ● | ● | ● | ○ | ○ | ○ | ● | ● | ● |
| Focus on key factors | ◐ | ● | ◐ | ● | ◐ | ◐ | ○ | ○ | ● | ○ | ● | ● |
| Determination of an AI deployment capability | ○ | ◐ | ○ | ◐ | ● | ● | ◐ | ● | ○ | ◐ | ○ | ● |
| Weighting of maturity indicators | ○ | ◐ | ○ | ◐ | ○ | ○ | ○ | ○ | ○ | ○ | ○ | ● |
| Derivation of suggestions for improvement | ○ | ○ | ● | ● | ● | ● | ◐ | ● | ● | ● | ● | ● |
| Industry-specific adaptation | ◐ | ● | ○ | ○ | ◐ | ○ | ◐ | ● | ● | ● | ◐ | ● |
| Support from top management | ◐ | ◐ | ○ | ○ | ● | ● | ◐ | ○ | ◐ | ○ | ○ | ● |
| Standards for data quality | ◐ | ◐ | ◐ | ○ | ○ | ◐ | ◐ | ○ | ○ | ● | ◐ | ● |
| Assessment of security-relevant aspects and data protection | ● | ● | ◐ | ● | ○ | ◐ | ○ | ○ | ● | ● | ◐ | ● |
| Consideration of the competition | ◐ | ● | ○ | ◐ | ○ | ◐ | ○ | ○ | ● | ○ | ○ | ● |
| Consideration and controls of implementation costs | ◐ | ● | ◐ | ◐ | ○ | ◐ | ● | ○ | ○ | ○ | ◐ | ● |
| Integration of a feedback process | ○ | ● | ○ | ○ | ○ | ○ | ● | ○ | ○ | ○ | ○ | ● |

Legend:
○ : Not fulfilled
◐ : Partly fulfilled
● : Fulfills

By reviewing the evolution of MLOps and its relationship to traditional software development methods, the paper by Liang et al [23] proposed show ways to integrate the system into machine learning to solve the problems faced by existing MLOps and improve productivity.

In Appization™ @ NeuraHub™, a user selects an AI model from the marketplace. The API Gateway authenticates the user, verifies permissions and routes the model to the appropriate edge device. The model is then containerized, enabling it to run independently on the edge device.

**Model Deployment Workflow**

**Steps involved:**

1. User Browses Marketplace.
2. User Selects AI Model.
3. API Gateway Authenticates and Routes Request.
4. Model is Deployed on Edge Device.
5. Device Begins Inference.

**Real-Time Monitoring and Management**

Once deployed, Appization™ @ NeuraHub™ provides users with real-time monitoring and management capabilities. Performance metrics, such as processing speed and model accuracy, are accessible through the user interface, allowing for prompt adjustments.

**Security and Privacy Protocols**

Security is a critical focus within Appization™ @ NeuraHub™, as edge devices often handle sensitive data. Appization™ @ NeuraHub™ employs encrypted protocols for data transmission, role-based access control, and multi-layered authentication to secure AI model interactions.

## V. Comparison With Mobile App Store Model

The Appization™ @ NeuraHub™ model adapts several features from mobile app stores, tailored specifically to meet the needs of edge AI. Below is a detailed comparison:

**Table 3: Mobile App Store vs Appization™ @ NeuraHub™ Platform**

| Feature | Mobile App Store | Appization™ @ NeuraHub™ Platform |
|---|---|---|
| **Interface** | GUI-driven, intuitive | Web/mobile interface for AI model access |
| **App Updates** | Managed through the app store | Managed through the API Gateway |
| **User Reviews** | Ratings, reviews for guidance | Metrics, logs, and feedback on models |
| **In-App Purchases** | Paid upgrades | Add-ons for advanced AI model features |
| **User Permissions** | Personal data permissions | Role-based access, secure communications |

## VI. Challenges And Potential Solutions In Appization™ @ Neurahub™

**Limited Edge Device Resources**

Edge devices often have limited processing power and storage[18]. Appization™ @ NeuraHub™ addresses this through lightweight models and resource-optimized algorithms that minimize computational requirements.

- **Solution**: Compression techniques, hardware acceleration and model pruning.

**Security and Compliance Requirements**

Due to the sensitive nature of edge AI applications, compliance with privacy and security standards is critical [19].

The deployment of AI models on edge devices introduces unique privacy and security challenges[19,24]. To mitigate these risks, it's essential to adopt robust measures:

**Privacy:**

- Data Minimization: Collect and process only the necessary data to accomplish the task.
- Anonymization and Pseudonymization: Mask personal identifiers to protect user privacy.
- Secure Data Storage: Employ encryption techniques to safeguard sensitive data, both at rest and in transit.
- Privacy-Preserving AI**:** Utilize techniques like federated learning and differential privacy to train models without sharing raw data.

**Security:**

- Secure Boot and Firmware Updates: Ensure the integrity of the device's software and prevent unauthorized modifications.
- Secure Communication: Implement secure communication protocols to protect data transmission.
- Robust Access Control: Restrict access to sensitive components and data.
- Regular Security Audits and Updates**:** Conduct regular security assessments and apply timely patches to address vulnerabilities.
- Threat Modeling: Identify potential threats and vulnerabilities to develop effective security strategies.
- Hardware Security Modules (HSMs): Utilize HSMs to protect cryptographic keys and sensitive data.

By prioritizing privacy and security, one can unlock the full potential of edge AI while safeguarding user data and system integrity.

Appization™ @ NeuraHub™ 's framework incorporates data encryption and strict access control.

- Solution: Role-based access, secure APIs, and real-time threat monitoring.

**Developer Ecosystem and Adoption**

Kern et al [20] argue that the common software defects are a systemic problem, rooted in the fundamental structure of the developer ecosystem. This ecosystem includes all the tools, processes, and systems used by developers to create and deploy software, such as programming languages, libraries, frameworks, version control systems, build tools, deployment platforms, and their configurations. According to Myllärniemi et al [25] Instead of focusing solely on the API, the framework owner should consider all platform boundary resources: API, development tools and information, also the boundary resources should support developers' needs throughout the developer journey, from early adoption to continuous use.

Appization™ @ NeuraHub™ 's success depends on widespread developer adoption, particularly for AI model development.

- Solution: Providing SDKs, incentives and a revenue-sharing model to attract a wide range of developers.

## VII. Strategic Benefits And Market Potential

**Scalability and Flexibility**

The Appization™ @ NeuraHub™ model allows users to scale and customize AI capabilities as needed, offering adaptability similar to downloading new apps on mobile handsets.

**Commercial Viability**

With an expanding market for edge AI devices, Appization™ @ NeuraHub™ presents a commercial opportunity by creating a platform that generates revenue for both model developers and device manufacturers.

**Future Outlook**

Appization™ @ NeuraHub™ can evolve into a standardized protocol for deploying AI models on various edge devices, extending beyond cameras to encompass drones, IoT sensors and industrial robots.

## VIII. Conclusion

The Appization™ @ NeuraHub™ framework takes inspiration from the successful mobile app store model to revolutionize the deployment of AI functions on edge devices, such as IndoAI's Edge Cameras. It aims to create a highly adaptable, scalable, and customizable platform that empowers users to effortlessly install, manage, and update AI models on these specialized devices.

This paper details into the intricacies of the Appization™ @ NeuraHub™ platform, positioning it as a comprehensive ecosystem where AI models can be deployed, managed and updated independently. The framework introduces standardized protocols, a developer-friendly marketplace, and a robust microservices architecture that streamline the process of deploying AI models on edge devices. By overcoming existing challenges and scaling edge AI for specialized applications, Appization™ @ NeuraHub™ paves the way for a future where AI-powered solutions are readily accessible and easily deployable on edge devices.

The Appization™ @ NeuraHub™ model offers a compelling solution to the challenges associated with traditional AI deployment methods, particularly in terms of scalability, flexibility, and security. By creating an ecosystem where AI models can be easily installed, updated, and tailored to specific needs, the Appization™ @ NeuraHub™ framework empowers edge devices to become more adaptable and versatile. This approach fosters innovation among developers, enabling them to create and share a diverse range of AI applications. Moreover, by mirroring the successful model of mobile app stores, the Appization™ @ NeuraHub™ framework brings AI functionalities closer to end-users, making AI-powered experiences more accessible and user-friendly.

## References:


[1] Department Of Commerce, February 2023, Competition In The Mobile Application Ecosystem, Https://Www.Ntia.Gov/Sites/Default/Files/Publications/Mobileappecosystemreport.Pdf Https://Mobile-Magazine.Com/Articles/How-Edge-Computing-Ai-Are-Changing-Mobile-Technology

[2] Rahul C. Basole, Jürgen Karlajürgen Karla, Value Transformation In The Mobile Service Ecosystem: A Study Of App Store Emergence And Growth, Service Science 4(1):24-41, March 2012, Doi: 10.1287/Serv.1120.0004 Https://Blogs.Nvidia.Com › Blog › What-Is-Edge-Ai

[3] Edge Ai Camera Is Reshaping The Future Landscape , Security World Market, Https://Www.Securityworldmarket.Com › News ›

[4] Carlos Cob-Parro , Cristina Losada-Gutiérrez , Marta Marrón-Romera , Alfredo Gardel-Vicente, Ignacio Bravo-Muño ,Smart Video Surveillance System Based On Edge Computingantonio , Sensors 2021, 21, 2958. Https://Doi.Org/10.3390/S21092958 Https://Www.Mdpi.Com/Journal/Sensors Https://Fpgainsights.Com/Artificial-Intelligence/Ai-Powered-Edge-Computing/

[5] Oumayma Jouini, Kaouthar Sethom , Abdallah Namoun 3 , Nasser Aljohani 3 ,Meshari Huwaytim Alanazi 4, Mohammad N. Alanazi, A Survey Of Machine Learning In Edge Computing: Techniques, Frameworks, Applications, Issues, And Research Directions, June 2024technologies 12(6) Doi: 10.3390/Technologies12060081, Https://Www.Mdpi.Com/Journal/Technologies

[6] Defining App Stores: The Role Of Curated Marketplaces In ...Slingerjansen.Nl Https://Slingerjansen.Nl › Uploads › 2009/04

[7] What Is An Api Gateway?, Https://Www.Ibm.Com › Think › Topics › Api-Gateway

[8] What Is An Api Gateway?, Https://Www.Paloaltonetworks.Com › Cyberpedia › What...

[9] Https://Www.Diva-Portal.Org/Smash/Get/Diva2:1895929/Fulltext01.Pdf

[10] Yitong Wang , Jun Zhao, Mobile Edge Computing, Metaverse, 6g Wireless Communications, Artificial Intelligence, And Blockchain: Survey And Their Convergence, Yitong Wang School Of Computer Science And Engineering, Nanyang Technological University, Singapore, Https://Arxiv.Org/Pdf/2209.14147

[11] X. Ran, H. Chen, X. Zhu, Z. Liu And J. Chen, Deepdecision: A Mobile Deep Learning Framework For Edge Video Analytics, Ieee Infocom 2018 - Ieee Conference On Computer Communications, Honolulu, Hi, Usa, 2018, Pp. 1421-1429, Doi: 10.1109/Infocom.2018.8485905.

[12] Mohan Harish, Srikar Podicheti, Deepak Kumar, Optimizing Energy Efficiency In Edge-Computing Environments With Dynamic Resource Allocation, International Journal Of Science And Engineering Applications Volume 13-Issue 07, 01 – 08, 2024, Issn:- 2319 - 7560doi: 10.7753/Ijsea1307.1001

[13] Rongxu Xu, Wenquan Jin, Dohyeun Kim, Microservice Security Agent Based On Api Gatewayin Edge Computing, Sensors 2019, 19, 4905; Doi:10.3390/S19224905 Www.Mdpi.Com/Journal/Sensors

[14] Nuno Mateus-Coelho , Manuela Cruz-Cunha , Luis Gonzaga Ferreira , Security In Microservices Architectures Panel Https://Www.Accenture.Com/Us-En/Insights/Cloud/Edge-Computing-Index Https://Arxiv.Org/Html/2410.05349v1

[15] Christoph Kern, Developer Ecosystems For Software Safety, Communications Of The Acm, Volume 67, Issue 6, Pages 52 – 60 , Https://Doi.Org/10.1145/3651621

[16] Sonntag, M., Mehmann, S., Mehmann, J., Teuteberg, F. (2024).Development And Evaluation Of A Maturity Model For Ai Deployment Capability Ofmanufacturing Companies, Information Systems Management, 1–31 Https://Doi.Org/10.1080/10580530.2024.2319041

[17] Ai Model Deployment , Https://Learn.Microsoft.Com › Ai › Playbook › Capabilities

[18] Penghao Liang, Bo Song ,Xiaoan Zhan ,Zhou Chen, Jiaqiang Yuan, Automating The Training And Deployment Of Models In Mlops By Integrating Systems With Machine Learning, Https://Arxiv.Org/Pdf/2405.09819

[19] Raghubir Singh, Sukhpal Singh Gill,Edge Ai: A Survey,Internet Of Things And Cyber-Physical Systems, Volume 3,2023,Pages 71-92, Https://Doi.Org/10.1016/J.Iotcps.2023.02.004.

[20] Myllärniemi, V., Kujala, S., Raatikainen, M. Et Al. Development As A Journey: Factors Supporting The Adoption And Use Of Software Frameworks. J Softw Eng Res Dev 6, 6 (2018). Https://Doi.Org/10.1186/S40411-018-0050-8