TV Boxes as Support Servers for IoT Applications

Tiago Godoi Bannwart, Gustavo P. C. P da Luz,
Gabriel Massuyoshi Sato, Luis Fernando Gomez Gonzalez, Juliana Freitag Borin
Instituto de Computação, UNICAMP, Campinas, Brazil
{*t215386, g271582, g172278*}*@dac.unicamp.br*
{*gonzalez, jufborin*}*@unicamp.br*

***Abstract*—This paper presents a fog computing server designed for Internet of Things (IoT) applications, which promotes the reuse of seized TV Box devices. A smart parking system deployed at the Institute of Computing, Unicamp served as the first IoT application to use the server with a TV Box model with limited hardware resources. The device was configured to provide monitoring services, enable remote firmware updates, and generate temporal metrics. Experimental results show that the system achieved operational stability, low energy consumption, and satisfactory performance within the proposed scenario. These findings confirm the technical feasibility of repurposing TV Boxes and highlight their potential as a low-cost and environmentally sustainable alternative for fog computing applications, in line with the principles of the circular economy.**

***Index Terms*—Internet of Things, Device Repurposing, Fog Computing, Smart Cities, TV Box**

I. Introduction

The increasing volume of seized illegal electronic devices in Brazil represents a significant logistical, economic, and environmental challenge [1]. Among these devices, TV Boxes stand out, confiscated in large quantities by the Federal Revenue Service (referred to as Receita Federal - RFB) due to being prohibited for lacking proper homologation. It is estimated that millions of units are stored in RFB warehouses, with a 2024 report indicating an approximate stock of 2.5 million devices [2].

Historically, the primary destination for these seized goods was total destruction. This process, in addition to involving substantial costs for the public sector, results in the generation of a considerable amount of electronic waste, which in turn requires proper disposal. In past operations, more than 97,000 pirated TV Box devices were destroyed by the RFB in partnership with the Brazilian Pay Television Association (Associação Brasileira de Televisão por Assinatura - ABTA) [3].

In recent years, however, a strategic shift has occurred driven by the pursuit of greater efficiency and sustainability. In line with the principles of the circular economy, which aim to minimize waste and maximize resource use efficiency [1], the RFB has established partnerships with educational and research institutions to explore more appropriate ways of utilizing seized materials. This initiative seeks to prevent the generation of electronic waste and provide a suitable destination for the equipment [2]. Consequently, the practice enables the availability of technological devices at a reduced cost, eliminating the financial burden of acquiring new equipment.

Some studies have successfully reused TV Boxes in innovative ways, such as Campos et al. [4], who used TV Boxes as educational platforms in schools, keeping Android and installing educational games. Pinhao Sobrinho et al. [5], on the other hand, compared the use of a TV Box as a personal computer with a TV Box as a Thin Client through the Remote Desktop Protocol (RDP), enabling access to a remote server with greater processing capacity.

Furthermore, TV Boxes have been used as edge devices running image-based Artificial Intelligence (AI) applications for people counting [6], [7], fall detection [8], and even for detecting diseases in leaves [9]. In the context of performing AI model inference at the edge, da Luz et al. [1] demonstrated the performance capabilities of TV Boxes through a stress test and showed that their use allows for a lower carbon footprint compared to traditionally used commercial devices.

Within this context of seeking new utilities and applications that mitigate waste, this article explores the feasibility of a practical application of high relevance in the current technological scenario, which, to the authors' knowledge, is being conducted for the first time. Specifically, it investigates the reuse of seized TV Boxes as fog computing devices, based on a Linux system that functions as a support server for Internet of Things (IoT) applications. This study demonstrates that these reused devices exhibit resilience and robustness, positioning them as a low-cost and energy-efficient alternative compared to commercially available hardware.

II. Case Study

The application that first used the proposed server was the smart parking system of the Institute of Computing (IC) at UNICAMP.

The following section describes the system's operation, the implemented infrastructure, and the challenges addressed through the use of a TV Box as a support server for IoT applications. Fig. 1 illustrates the simplified dynamics of the original system.

The central objective of the smart parking project is to identify the number of available parking spaces and provide this information on a totem with LED displays located at its entrance. To perform the counting, a camera connected to a Raspberry Pi is positioned to capture aerial images of the entire parking area. From these images, Deep Learning models identify the number of vehicles present. This value is then subtracted from the total number of spaces, resulting in

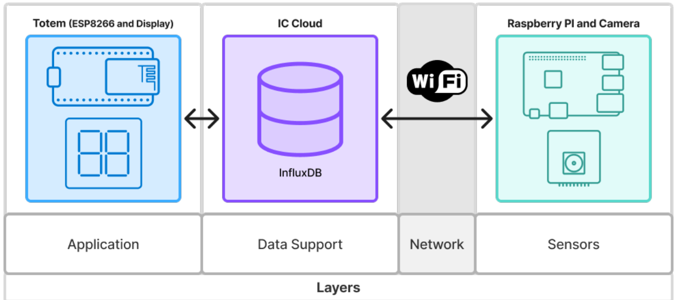

Fig. 1. Simplified Dynamics of the previous Smart Parking System.

the number of free spots. Once calculated, this information is recorded in the InfluxDB time-series database, allowing other applications to consume it, notably the display totem.

The totem was built with LED strips configured as seven-segment displays, controlled by an ESP8266 microcontroller. Its function is to retrieve, from the database, the current number of available spaces and display it on the panel in real time. More details can be found in a previous work [10], which explains the operation of each component of the system and the accuracy of the model that counts the available spots.

Previously, the data was sent directly from the database to the totem, which caused maintenance challenges and motivated the adoption of a higher-level server to support the application. With the introduction of the TV Box, it became possible to supply data to the totem without connecting it directly to the database, while also enabling remote firmware updates. In addition, the solution made it feasible to monitor the status and performance of both the totem's microcontroller and the Raspberry Pi. Another requirement addressed by this approach was the creation of an environment that supports the generation of metrics, numerical analyses, and graphical visualizations related to parking lot usage.

III. Methodology

A. Replacement of the Native Operating System

The system was developed using the TV Box model called Youit TX2. This device is based on hardware with limited specifications, including a 32-bit ARM Cortex-A7MP CPU with four cores operating at 1.2 GHz. The system features 2 GB of LPDDR3 RAM and 16 GB of eMMC flash storage. The device also supports Ethernet, WiFi, and Bluetooth. Due to its computational power, the Youit TX2 is comparable in processing capability to a Raspberry Pi 3B [1].

The initial stage of the project consisted of replacing the original Android operating system with a Linux distribution, adopting Armbian, a Debian-based variant optimized for ARM hardware. This distribution is open source, customizable, and resource-efficient[1]. The version used was 25.8.0 Bookworm Minimal, which does not include a desktop environment, chosen for its lightweight nature and to meet the project requirements. We also tested a newer version of Armbian on a second device to analyze how the system would behave.

To write the image to the internal flash memory, the Multi-tool is used, which is a small but powerful tool designed for flashing images and general system maintenance, made available on the Armbian community forum[2]. After completing the flashing and rebooting the system, the device boots into the new operating system, prompting for basic configurations.

B. Pre-configuration Tool

After the first boot, the system executes a set of routines, called *First Login*, that are responsible for the initial configuration of the device. However, by default, this process occurs manually and therefore becomes a bottleneck when multiple TV Boxes need to be configured. To overcome this limitation, a bash script was developed capable of performing a pre-configuration on the image before flashing, making the device ready for use and remotely accessible from the first boot.

The script works by including a specific configuration file, in the '/root' directory within the image, containing configuration directives. According to the official documentation, to achieve a fully autonomous configuration, all entries in this file must be completed, which are then interpreted by the *First Login* routines. Additionally, Armbian allows the creation of a script, located in the same directory, that runs after the *First Login* is completed. This feature can be used for supplementary tasks, such as package installation, system adjustments, or service enabling[3]. In this study, this functionality was used to install and activate the Tailscale service[4], which connects the device to a simplified VPN, forming a secure private network among participating nodes and, in practice, enabling remote access from any network.

However, the Getty service requests authentication with the default user and password credentials on the first boot. To overcome this restriction, the pre-configuration script creates an override in the Getty settings, enabling autologin for the root user. This mechanism, while necessary to automate the initial step, must be disabled later for security reasons.

C. Architecture

To meet the needs described above, seven services were configured on the TV Box, as illustrated in Fig. 2, which operate in parallel. Among them, five were developed specifically for this case study, while the other two, NGINX and Tailscale Funnel, correspond to already established solutions. Additionally, the device was configured to reboot every six hours to enhance system stability and reliability.

1) Totem Services: Two services were implemented to support the totem:

- **Parking Lot API**: Provides a web endpoint with the current number of available spaces. Also implements a heartbeat mechanism, where the ESP8266 sends an

[1] Armbian Project

[2] Armbian Forum Thread about this type of device
[3] Armbian Documentation
[4] How Tailscale Works

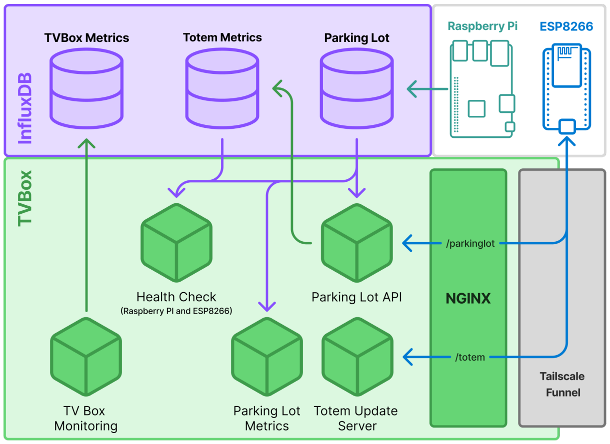

Fig. 2. Architecture being executed on the TV Box.

identifier and available heap in HTTP headers to monitor proper operation.

- **Totem Update Server**: Serves as a firmware repository for the ESP8266, enabling periodic autonomous updates via HTTP. To facilitate this process, a CLI (Command Line Interface) tool automates version insertion, compilation, and upload, allowing remote totem updates and simplifying maintenance.

The three services described are managed by SystemD, which collects logs and ensures automatic recovery in case of failures or reboots. To enable access to the two web servers from outside the local network without administrative permissions, the Tailscale Funnel[5] was employed as a tunneling mechanism, exposing the HTTPS port through its servers. NGINX[6] was installed as a reverse proxy to properly route incoming traffic from the Funnel based on the requested URL.

2) Health Check: Two scripts are executed every 10 minutes to monitor the health of the smart parking devices:

- **Raspberry Pi Status**: Checks the last timestamp sent by the Raspberry Pi to the InfluxDB bucket with the parking count.
- **Totem Status**: Queries another InfluxDB bucket containing the heartbeat from the ESP8266 in the totem. This allows verification that the device is active and has processed the latest available parking count data.

If more than five minutes pass without updates, the system considers a communication failure and sends an alert via Telegram, indicating whether the problem occurred with the Raspberry Pi or the totem.

3) Parking Lot Metrics: Two additional scripts are responsible for generating periodic statistics and reports on parking lot usage:

- **Daily Statistics**: Daily text-based metrics report the previous day's parking usage, comparing it with the 30-day historical average. During testing, this process was executed every hour to simulate daily reporting.
- **Weekly Reports**: A weekly graphical dashboard shows overall occupancy for the current week, using the last 40 days of data for comparison. During testing, this process was executed daily to simulate the weekly reporting.

The results produced by both scripts are sent by the TV Box to the responsible users via Telegram, facilitating remote analysis of the operation.

IV. Results

The results presented aim to evaluate the performance of the TV Boxes during the testing period, as well as their energy consumption. The evaluation of the proposed system's performance was conducted based on data collected from two TV Boxes of the same model over a seven-day period. Table I provides a summary of the results obtained.

It can be observed that the TV Boxes exhibited low CPU and memory usage while maintaining a low CPU temperature. TV Box 1 showed lower performance compared to TV Box 2. One possible explanation is the presence of a camera connected to the USB port of TV Box 1, even though it was not used to capture images during this period. This may have caused the device to consume more power and generate more heat. The higher memory usage on TV Box 1 could also be attributed to the two TV Boxes running on different Armbian versions.

The energy consumption data of the two TV Boxes were collected using an INA219 module connected to each device's power supply. In this way, the instantaneous power were recorded every minute over a one-week experiment. An ESP8266 was used to send the data to a bucket in InfluxDB for calculating the average consumption during the operation of the TV Boxes. It is observed that the devices exhibit low energy consumption, equivalent to approximately 12 kWh to 13 kWh per year. A 10 W LED lamp consumes 87.6 kWh in a year, using roughly seven times more energy compared to a TV Box.

Aiming to analyze one of the TV Boxes' metrics to understand the server's behavior, memory was chosen as it is a limited resource that directly reflects the impact of running services. Fig. 3 shows the memory consumption of each device over time.

Some patterns can be observed in Fig. 3. The first occurs when memory usage exceeds the average. One of these peaks, marked in red, corresponds to the execution of the code that runs once a day to send an image via Telegram. The second, highlighted in yellow, is associated with the process executed hourly to send a text message.

In addition to these peaks, approximately every six hours there is a drop in average memory usage, corresponding to the period when the devices perform a reboot. This behavior

TABLE I
Statistical Summary of Seven Days of Device Analysis Metrics

Device	*samples*	*CPU (%)*	*Memory (MB)*	*CPU Temp. (°C)*	*Power (W)*
tvbox-01	9993	2.72 ± 4.76	183.62 ± 15.35	60.19 ± 1.55	1.54 ± 0.10
tvbox-02	10024	2.41 ± 4.69	159.32 ± 14.73	55.19 ± 1.40	1.38 ± 0.09

[5] Tailscale Funnel Service
[6] NGINX Project

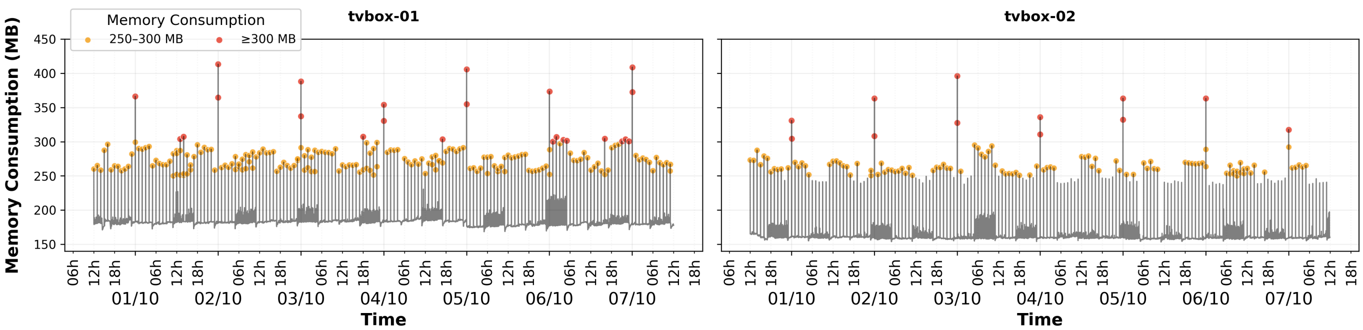

Fig. 3. Memory consumption of each TV Box through time. Peaks are highlighted.

demonstrates that the restart strategy is effective as a preventive measure to maintain device stability and avoid overloads.

It is worth noting that the use of TV Boxes as support servers for IoT applications showed lower baseline CPU and memory usage compared to using the same device for image processing with AI models [1], [7].

Finally, the collection of ESP8266 heartbeats, enabled by the TV Box, revealed that the microcontroller maintains a regular amount of free heap during operation, indicating that the totem operates stably.

V. Conclusion and Future Work

This work proposed the reuse of TV Boxes as support servers for IoT applications in the smart parking system at the Institute of Computing, Unicamp. The Youit TX2 demonstrated stable operation, low energy consumption, and adequate performance, fulfilling all requirements. The study also highlights the potential of such platforms to support the circular economy by providing a sustainable, low-cost alternative for seized devices. Future work may explore applying this architecture to other TV Box models and IoT applications.

It is also possible to expand the architecture to support multiple parking lots, a scenario that would likely require more device resources. To increase system reliability, there is also the proposal to develop a load balancer coupled to multiple TV Boxes to distribute requests, eliminating single points of failure and enhancing system resilience and availability.

Regarding system health monitoring, this study used alert message services via Telegram targeted at a single user. Future trials could be conducted to evaluate the scalability of the solution, considering support for multiple simultaneous users.

The code for the Armbian image pre-configuration tool and the other services presented in this project, along with usage instructions, is publicly available on GitHub[7].

Funding

This research was supported by CAPES (process 88887.189925/2025-00), by the Brazilian Ministry of Science, Technology and Innovations, with resources from Law num. 8,248, of October 23, 1991, within the scope of PPI-SOFTEX, coordinated by Softex and published Arquitetura Cognitiva (Phase 3), DOU 01245.003479/2024-10.

[7]GitHub repository

References

[1] G. P. da Luz, G. M. Sato, L. F. G. Gonzalez, and J. F. Borin, "Repurposing of tv boxes for a circular economy in smart cities applications," *Scientific Reports*, vol. 15, no. 1, p. 22638, 2025, doi:10.1038/s41598-025-97379-4.

[2] R. Jansen. (2024) Cigarro, bebida e aparelho de 'gatonet': como universidades usam apreensões da receita na pesquisa. Acessed in: 7 out. 2025. [Online]. Available: https://perma.cc/SF8B-DHC3

[3] Secretaria-Geral da Presidência da República. (2021) Receita federal e associação brasileira de televisão por assinatura destroem mais de 97 mil aparelhos de tv box piratas no rio de janeiro. [Online]. Available: https://perma.cc/DC3J-FNF5

[4] R. A. Campos, F. A. N. Martini, J. T. F. Suzuki *et al.*, "Projeto recriar: sustentabilidade, inclusão digital e inovação pedagógica a partir da reutilização de dispositivos tecnológicos ilegais," *Editora Impacto Científico*, pp. 2100–2121, 2025, doi:10.56238/arev7n7-076.

[5] M. P. Sobrinho, J. Sousa, F. Carvalho, C. Silva, and L. Santos, "Explorando tv boxes em ambientes educacionais: Vantagens e limitações operando como thin client," in *Anais da XII Escola Regional de Computação do Ceará, Maranhão e Piauí*, 2024, pp. 249–258, doi:10.5753/ercemapi.2024.243796.

[6] G. Sato, G. Luz, L. Gonzalez, and J. Borin, "Reaproveitamento de tv boxes para aplicação de contagem de pessoas na borda em cidades inteligentes," in *Anais do VIII Workshop de Computação Urbana*. Porto Alegre, RS, Brasil: SBC, 2024, pp. 197–209, doi:10.5753/courb.2024.3375.

[7] G. M. Sato, G. P. da Luz, L. F. G. Gonzalez, and J. F. Borin, "People counting application with crowded scenarios: A case study with tv boxes as edge devices," *Journal of Internet Services and Applications*, vol. 16, no. 1, pp. 470–479, 2025, doi:10.5753/jisa.2025.5156.

[8] F. Shu and J. Shu, "An eight-camera fall detection system using human fall pattern recognition via machine learning by a low-cost android box," *Scientific Reports*, vol. 11, no. 1, p. 2471, 2021, doi:10.1038/s41598-021-81115-9.

[9] R. Moreira, L. F. R. Moreira, P. L. A. Munhoz, E. A. Lopes, and R. A. A. Ruas, "Agrolens: A low-cost and green-friendly smart farm architecture to support real-time leaf disease diagnostics," *Internet of Things*, vol. 19, p. 100570, 2022, doi:10.1016/j.iot.2022.100570.

[10] G. P. da Luz, G. M. Sato, T. G. Bannwart, L. F. G. Gonzalez, and J. F. Borin, "10 years of deep learning for vehicle detection at a smart parking: What has changed?" in *Workshop de Computação Urbana (CoUrb)*. SBC, 2025, pp. 127–140, doi:10.5753/courb.2025.8869.