Integration of Reusable Optical Media into Home Cinema via Raspberry Pi

Diana Feusi

Fachhochschule Graubünden
BSc Computational and Data Science

E-Mail: `diana.feusi@stud.fhgr.ch`

December 17, 2025

Abstract

This thesis presents a comprehensive workflow and system architecture for a cost-effective and legally compliant home cinema platform, utilizing the LibreELEC distribution on a Raspberry Pi 4 single-board computer. The work details the deployment of the Kodi-based media center and the systematic digitization of legacy DVD media into optimized MP4 containers. A central component of the implementation is a custom-developed Python script that automates metadata retrieval and artwork integration via the The Movie Database (TMDB) API, ensuring a user experience comparable to commercial streaming services.

Furthermore, the system's interoperability is enhanced through the integration of mobile remote control applications, bridging the gap between local physical media and modern digital usability. The final evaluation confirms the Raspberry Pi 4's high performance in high-definition playback and highlights the system's energy efficiency and sustainability. This work demonstrates that open-source software and low-cost hardware can effectively serve as a versatile, self-hosted alternative to proprietary media ecosystems.

1 Introduction

This work presents a cost-effective and legally compliant home cinema platform based on the Raspberry Pi 4. The system utilizes the LibreELEC distribution in conjunction with the Kodi media center to facilitate the playback of physical DVD media that has been digitized into the MP4 format. By transforming legacy DVDs into a format compatible with single-board computers (SBCs), users can establish a private, reusable media library that offers the convenience of modern streaming without dependence on external subscription-based services.

The primary motivation for this system is the revitalization of existing home media collections. As optical drives become increasingly rare in modern laptops and consumer electronics, extensive DVD collections often remain unused. Furthermore, the rising costs and fragmenting content of commercial streaming platforms provide a strong economic incentive for self-hosted solutions. By converting these physical assets into digital files stored on high-capacity external drives, old media becomes practically accessible again, achieving a level of usability comparable to commercial alternatives. The objective of this thesis is to design, implement, and evaluate this architecture, focusing on performance, automation, and system stability.

2 State of Research

Media center solutions for the Raspberry Pi are predominantly based on specialized Linux distributions. These systems are engineered to store, stream, and manage multimedia data efficiently. Most distributions utilize *Kodi* as the core software layer, which provides a sophisticated graphical user interface (GUI) for media management. Two prominent examples in this domain are LibreELEC and OSMC (Open Source Media Center).

LibreELEC follows the "Just enough OS" philosophy, offering a lightweight and highly stable environment that is ideal for dedicated media appliances. However, its read-only file system makes the installation of third-party Linux applications outside the Kodi repository more complex. In contrast, OSMC is built upon a full Debian-Linux distribution, offering greater flexibility for users who require a complete operating system environment for background tasks (**Betriebsystem**; Elektronik-Kompendium, 2025).

Compared to proprietary streaming services like Netflix or Disney+, open-source media centers offer superior customization. While commercial services limit users to licensed content, a Raspberry Pi-based system can aggregate local files, network storage, and various streaming plugins into a single, unified interface.

To integrate physical media into this digital ecosystem, the content must undergo a conversion process. For non-encrypted DVDs, this involves direct transcoding into digital containers HandBrake Documentation, 2025. However, encrypted media requires a "ripping" process to bypass Digital Rights Management (DRM) and extract the raw data into unprotected formats DVD.net.au, 2025. Current research highlights that the primary differences between available tools lie in their processing speed, hardware-accelerated encoding support, and the resulting output quality.

Recent literature emphasizes the versatility of the Raspberry Pi 4 in multimedia applications due to its support for hardware-accelerated 4K decoding Mathe and Kondaveeti, 2024. Technical evaluations of LibreELEC and OSMC confirm their optimization for SBC architectures (Follmann, 2018), while studies on the Kodi software architecture illustrate how open-source frameworks can be adapted for high-performance multimedia tasks (Bromley et al., 2020). For the transcoding stage, open-source tools such as HandBrake remain the industry standard for converting physical media into efficient digital formats like H.264 or H.265 (**PraxistippsHandBrake**).

3 System Overview and Architecture

The system is designed to implement a robust home media center based on the Raspberry Pi 4 platform, facilitating both the integration of modern streaming services and the playback of legacy DVD media. The architecture is composed of the Raspberry Pi hardware, the LibreELEC operating system—a specialized distribution for the Kodi media center—and various external peripherals, including high-capacity USB storage and optical drives.

A high-level conceptual view of the system architecture and its hardware-software interaction is provided in Figure 1. This diagram illustrates the primary hardware components and their functional data paths, encompassing the central processing unit, storage integration via high-speed interfaces, the media conversion workflow, and the final audiovisual output stage to the television.

As depicted in Figure 1, a Raspberry Pi 4 Model B serves as the primary processing unit. It is interfaced with an external 4 TB HDD via a USB 3.0 port to ensure sufficient bandwidth for high-capacity media storage and retrieval. Audio and video signals are transmitted via HDMI to a WiZ HDMI Sync Box, which subsequently routes the output to a 75-inch Samsung LED TV. Each component is powered by its respective power supply: the Raspberry Pi requires a 5 V/3 A DC input, the HDMI Sync Box operates at 12 V DC, and the television is connected to a standard 220–240 V AC power source.

3.1 Hardware and Software Components

The hardware core consists of the Raspberry Pi 4 Model B, equipped with 4 GB of LPDDR4 RAM. The device features a versatile I/O layout, including two Micro-HDMI ports, four USB ports (two USB 3.0 and two USB 2.0), a Gigabit Ethernet interface, and a MicroSD slot for the operating system.

The system runs LibreELEC (version 12.2.1), a "Just enough OS" Linux distribution specifically optimized for Kodi (version 6.12.56). Kodi serves as the primary application layer, providing a sophisticated graphical user interface (GUI) for navigation, local media playback, and plugin-based streaming. While LibreELEC is compatible with Raspberry Pi models 3, 4, and 5, the more recent generations (4 and 5) are recommended for 4K playback due to their superior hardware-accelerated decoding capabilities. The official hardware compatibility documentation is available at libreelec.tv.

The Raspberry Pi 4 is particularly well-suited for this media platform as it supports UHD resolution (3840x2160 pixels). In comparison, achieving stable playback and high refresh rates on older hardware like the Raspberry Pi 3 requires significant configuration effort and often results in lower performance Raspberry Pi StackExchange, 2025.

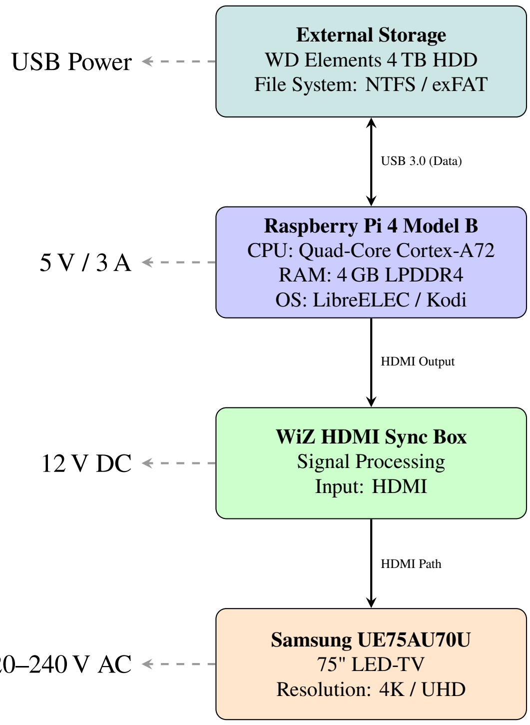

Figure 1: Hardware architecture and connectivity of the media center system.

LibreELEC is compatible with Raspberry Pi models 3, 4, and 5; however, models 4 and 5 are recommended for optimal 4K playback performance. The official compatibility list is available online at https://libreelec.tv/downloads/raspberry/. Network protocols such as HTTP (HyperText Transfer Protocol)

and RTSP (Real-Time Streaming Protocol) are theoretically supported, although RTSP was not explicitly tested in this work. Applications such as Red Bull TV and YouTube were tested and functioned smoothly without perceptible interruptions.
The Raspberry Pi 4 is well-suited for this media platform because it supports 4K resolution (3840x2160 pixels). While the older generation, such as the Raspberry Pi 3, is capable of media playback, achieving stable 4K resolution and a consistent refresh rate (Hz level) requires significantly more effort and is not as reliable as with the Raspberry Pi 4 or 5. Further user experience details and comparisons can be found in reference Raspberry Pi StackExchange, 2025.

3.2 Software Installation

The installation of LibreELEC was performed using a 32 GB MicroSD card. To prepare the boot medium, the Raspberry Pi Imager (v1.9.6) was utilized on a host computer. Within the imager, the correct hardware model was selected, and the LibreELEC image was sourced via the "Media player OS" menu. After the flashing process, the card was inserted into the Pi. Upon the initial boot, the system prompted for network configuration (supporting both 2.4 GHz and 5 GHz Wi-Fi bands), after which the media center was fully operational.
A primary advantage of LibreELEC over general-purpose distributions is its optimized performance; since no unnecessary background processes are active, the system maintains a low resource footprint. Furthermore, its read-only file system architecture ensures high stability and protection against accidental configuration errors. In contrast, OSMC (Open Source Media Center) is based on a full Debian-Linux system. While OSMC offers greater flexibility and a dedicated App Store for GUI-based installations, it inherently consumes more system resources due to the overhead of a complete operating system running in the background Fromaget, 2025.

3.3 Media Conversion and Playback

To enable the playback of legacy DVD media, physical content must be converted into digital formats. Two distinct approaches were utilized based on the disc's encryption:

- **Non-copy-protected DVDs:** These were directly transcoded to the MP4 container format using HandBrake HandBrake Documentation, 2025. To maintain a balance between quality and file size, the video encoder was set to H.264 (x264) with a constant Quality (RF) value of 20 and a frame rate of 30 fps.

- **Copy-protected DVDs:** For encrypted media, an intermediate step was required. The content was first decrypted and extracted into a Matroska (MKV) container using MakeMKV MakeMKV, 2025 and a valid Beta-Key **KeyMakeMVK**. This allowed for the selection of specific audio tracks and languages to reduce storage overhead. The resulting MKV file was then transcoded to MP4 following the same parameters as non-protected media.

Proper organization is crucial for automated metadata retrieval in Kodi. Files are named following a strict convention: *Movie Title (Year)*. This allows the system to scrape cover art and descriptions from online databases The Movie Database, 2025. This process was further optimized through a custom Python script (see Appendix) that automates the naming and folder structure organization.

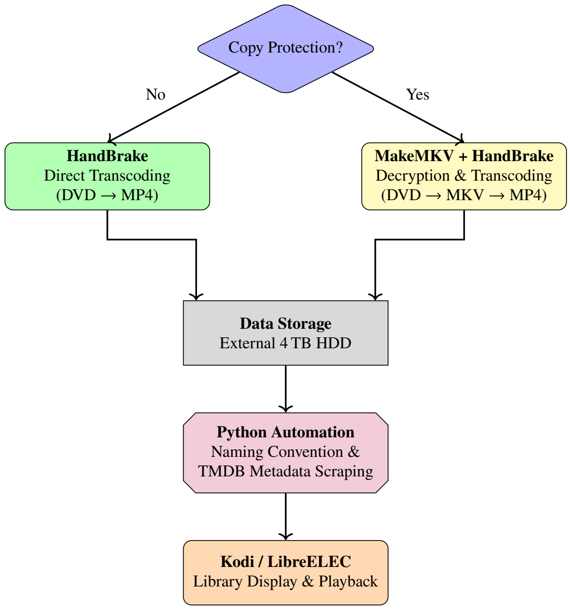

Figure 2: Process workflow: From physical media conversion to automated library management.

3.4 System Interaction

The interaction between system components is summarized as follows:

1. Media content is provided via external hard drives.

2. Conversion software processes raw content into Kodi-compatible formats (MP4).

3. The Raspberry Pi running LibreELEC hosts Kodi, which reads the media from storage and manages playback.

4. Audio and video output are delivered to connected displays and speakers.

5. Network streaming is handled directly by Kodi when internet connectivity is available.

Table 1: Technical specifications of sample converted media files

Movie Title (Year)	Format	Duration (HH:MM:SS)	Video Bitrate [kbit/s]	Overall Bitrate [kbit/s]	Size [GB]
Intouchables (2011)	MP4	01:47:30	1’076	1’686	1.27
The Devil Wears Prada (2006)	MP4	01:49:30	1’020	1’600	1.29
Training Day (2001)	MP4	01:57:04	998	1’544	1.26
Lord of the Rings III (2003)	MP4	03:12:47	1’496	2’106	2.84
Erkan und Stefan (2000)	MP4	01:23:28	1’147	1’310	0.78

3.5 Controlling Kodi

LibreELEC can be operated using a mouse and keyboard, typically during installation and initial setup. For daily use, however, this input method is inconvenient. To improve usability, the *Official Kodi Remote* app (Version 1.18.1) was installed.

The application provides full-featured remote control for the media center. To connect the app to LibreELEC, the server port and the MAC address must be configured. HTTP access must also be enabled in the LibreELEC system settings. Once configured, the app allows seamless control of Kodi from a smartphone or tablet.

3.6 Legal Considerations

In Switzerland, it is legally permissible to convert DVDs for personal use. Copying or distributing content to third parties remains illegal. This distinction is important when handling copy-protected media Schweizerischer Bundesrat, 2025.

3.7 Performance Considerations

The evaluation of playback quality confirmed that the Raspberry Pi 4 handles HD content without perceptible latency. During testing, the Kodi diagnostic overlay (shortcut `i`) was used to monitor the stream telemetry. For the film *Memoirs of a Geisha*, a playback resolution of 480p was observed. It is important to emphasize that this is a limitation of the source DVD’s native resolution and not a bottleneck caused by the Raspberry Pi 4 hardware or the LibreELEC software.

The computationally intensive transcoding tasks were conducted on an Intel i7 laptop. During these tests, CPU utilization consistently reached 80–90%, while RAM usage peaked at 9.8 GB in total (61% utilization). These metrics validate that while the Pi is an excellent playback device, the initial conversion process requires the processing power of a modern workstation. As shown in Table 1, the average file size is approximately 2.3 GB per movie. Consequently, a 4 TB HDD can store approximately 1’739 titles.

3.8 Automated Library Management via Python

To maintain a professional media library, consistent metadata and a structured directory hierarchy are required. A custom Python script was developed to automate the process of renaming files and retrieving metadata. This section describes the algorithmic logic and the integration with external services.

3.8.1 Algorithmic Workflow

The script follows a multi-stage execution flow to ensure high accuracy in movie identification:

1. **Directory Parsing:** The script scans the source directory for video containers (e.g., .mp4, .mkv).
2. **Pattern Matching:** The script extracts the title and the release year from the raw filename, which is often cluttered with encoding tags.
3. **API Interaction:** The script queries the *The Movie Database* (TMDB) API using the extracted title. It sends an asynchronous HTTP GET request and receives a JSON response containing the unique Movie-ID, official title, and poster URL.
4. **File Reorganization:** Once validated, the script moves the file into a dedicated folder named *Movie Title (Year)* and downloads the corresponding artwork as `poster.jpg` for Kodi to display.

3.8.2 API Integration and JSON Parsing

The interaction with the TMDB API is a core component. Below is a simplified representation of the metadata retrieval logic implemented in the script:
This automation significantly reduces the manual effort of library maintenance and ensures that Kodi’s "Scraper" can index the files with a 100% success rate.

3.9 Technical Evaluation: Codec Efficiency and Storage

The decision to use the H.264 (AVC) codec instead of the more modern H.265 (HEVC) for the DVD conversion was based on a trade-off analysis. While H.265 offers approximately 50% better compression for 4K

content, H.264 was chosen for the legacy DVD material (480p/576p) due to its universal compatibility and significantly lower transcoding time on the host workstation.
As shown in Table 1, the overall bitrate of approx. 1,600 kbit/s provides a transparent quality relative to the original DVD source while keeping the file size around 2.3 GB. This ensures that the 4 TB storage capacity remains sufficient for a collection of over 1,700 titles without the need for aggressive compression artifacts.

3.10 Sustainability and Energy Consumption

A key advantage of utilizing a Raspberry Pi 4 over a conventional x86-based Home Theater PC (HTPC) is energy efficiency.

- **Raspberry Pi 4:** Consumes approximately 3–5 W during idle and 7–10 W during 4K playback.
- **Standard Desktop PC:** Typically consumes between 80–150 W for similar tasks.

Assuming an average daily usage of 4 hours, the Raspberry Pi 4 consumes roughly 14.6 kWh per year. In contrast, a desktop PC would consume approximately 219 kWh. This reduction of over 90% in energy consumption highlights the project's contribution to sustainable computing.

4 Conclusion

In conclusion, the deployment of a Raspberry Pi 4 with LibreELEC and Kodi proved to be an effective and user-friendly solution for home media management. While the conversion of physical DVDs is time-consuming—especially when dealing with copy protection and multi-stage decryption—it represents a sustainable method for repurposing existing media collections.
Legal considerations are paramount: in Switzerland, ripping DVDs for personal use is permitted, though distribution remains strictly prohibited. By automating the library organization via Python and integrating streaming services like Disney+, the system achieves a level of functionality comparable to proprietary solutions. Ultimately, the performance analysis confirms that the Raspberry Pi 4 is a powerful playback unit whose output quality is primarily dictated by the source material rather than hardware constraints.

5 Discussion

The implementation reveals that while the Raspberry Pi 4 is a powerful playback platform, the integration of streaming services like Disney+ remains complex due to necessary DRM (Digital Rights Management) configurations (e.g., Widevine CDM). This highlights a trade-off between the flexibility of an open-source system and the "plug-and-play" nature of proprietary devices.
The DVD conversion process was identified as the primary bottleneck. Instabilities during ripping are largely attributed to copy-protection mechanisms, and the high data volume necessitates the use of external HDDs over standard USB flash drives. Furthermore, the visual quality is inherently limited by the standard-definition source material. While the system provides excellent upscaling, the low pixel density of legacy DVDs remains visible on large 4K displays.
From a legal perspective, the system operates within the boundaries of Swiss law for personal use, though it requires strict user responsibility regarding distribution. Performance testing showed no perceptible difference between LibreELEC and native Smart TV applications, confirming the efficiency of the Raspberry Pi 4.
Future improvements could enhance usability by transitioning to a Network Attached Storage (NAS) solution and fully implementing smartphone-based remote control via the Kodi JSON-RPC protocol. In conclusion, the system offers a sustainable and cost-effective alternative to commercial services, successfully bridging the gap between physical media archives and modern digital convenience.

Declaration of AI Usage

This report was prepared with the assistance of Artificial Intelligence (AI) tools. Specifically, Large Language Models (LLMs) were used for the following tasks:

- **Language Refinement:** Improving the linguistic flow and grammatical accuracy of the English text.
- **Technical Visualization:** Assistance in generating the TikZ code for the system architecture and workflow diagrams.
- **LaTeX Troubleshooting:** Support in debugging document structure and formatting issues.

All technical concepts, hardware selections, and the logical implementation of the Python script remain the original work of the author.

References

Bromley, S. T., Sheppard, J., Scanlon, M., & Le-Khac, N.-A. (2020). Retracing the flow of the stream: Investigating kodi streaming services. *arXiv preprint arXiv:2012.01107*.

DVD.net.au. (2025). *How to rip a dvd*. Retrieved December 14, 2025, from dvd.net.au

Elektronik-Kompendium. (2025). *Medienzentrum raspberry pi*. Retrieved October 1, 2025, from www.elektronik-kompendium.de

Follmann, R. (2018). *Das raspberry pi kompendium*. Springer Vieweg.

Fromaget, P. (2025). *Osmc vs libreelec: Doing more with these distributions*. Retrieved December 17, 2025, from raspberrytips.com

HandBrake Documentation. (2025). *Video conversion and transcoding*. Retrieved December 14, 2025, from handbrake.fr

MakeMKV. (2025). *Download makemkv for windows and linux*. Retrieved November 18, 2025, from www.makemkv.com

Mathe, S. E., & Kondaveeti, H. K. (2024). A comprehensive review on applications of raspberry pi. *Computer Science Review*, *51*, 100615.

Raspberry Pi StackExchange. (2025). *Can raspberry pi 3 do 4k video?* Retrieved December 17, 2025, from raspberrypi.stackexchange.com

Schweizerischer Bundesrat. (2025). *Bundesgesetz über das urheberrecht und verwandte schutzrechte*. Retrieved November 18, 2025, from www.fedlex.admin.ch

The Movie Database. (2025). *The movie database (tmdb) api documentation*. Retrieved December 14, 2025, from www.themoviedb.org

APPENDIX: Technical Implementation

.1 Python Script for Media Organization

```

import os
import re
import requests
from guessit import guessit

# Key und Dateinamen

API_KEY = "3a579b03de4384af4c5e39e54ce53576"
SOURCE_DIR = r"E:\Filme"
TARGET_DIR = r"E:\Filme"

CREATE_MOVIE_FOLDER = True  # True = Film in
    eigenen Ordner legen
LANGUAGE = "de-DE"          # de-DE oder en-US
DRY_RUN = False

VIDEO_EXTENSIONS = (".mkv", ".mp4", ".avi", ".mov
    ")

# ==============================
# FUNKTIONEN
# ==============================

def clean_filename(name):
return re.sub(r'[<>:"/\\|?*]', '', name)

def search_movie(title, year=None):
url = "https://api.themoviedb.org/3/search/movie"
params = {
        "api_key": API_KEY,
        "query": title,
        "language": LANGUAGE
}
if year:
params["year"] = year

r = requests.get(url, params=params)
r.raise_for_status()
results = r.json()["results"]
return results[0] if results else None


# HAUPTLOGIK

for file in os.listdir(SOURCE_DIR):
if not file.lower().endswith(VIDEO_EXTENSIONS):
continue

info = guessit(file)

title = info.get("title")
year = info.get("year")

if not title:
print(f" Titel nicht erkannt: {file}")
continue

movie = search_movie(title, year)
if not movie:
```

```
print(f" Film nicht gefunden: {file}")
continue

movie_title = movie["title"]
movie_year = movie.get("release_date", "0000")[:4]

clean_title = movie_title.replace(":", " -")

new_filename = f"{clean_title} ({movie_year}){os.path.splitext(file)[1]}"

if CREATE_MOVIE_FOLDER:
target_path = os.path.join(TARGET_DIR, f"{clean_title} ({movie_year})")
else:
target_path = TARGET_DIR

os.makedirs(target_path, exist_ok=True)

old_path = os.path.join(SOURCE_DIR, file)
new_path = os.path.join(target_path, new_filename)

if DRY_RUN:
print(f"[TEST] {old_path}  {new_path}")
else:
os.rename(old_path, new_path)
print(f"{file}  {new_filename}")
```

Listing 1: Python script for organizing DVD media files