

# Understanding LibreELEC and Kodi

## What is LibreELEC?

LibreELEC (Libre Embedded Linux Entertainment Center) is a **lightweight, open-source operating system** designed specifically to run media-center software like **Kodi** on small devices such as the **Raspberry Pi**, **Intel NUC**, or other low-power hardware.

Think of LibreELEC as:

“A minimal Linux system that boots directly into a full-screen media player.”

It's optimized for:

- **Speed and stability** (boots in under 15 seconds),
- **Media playback** (hardware-accelerated video/audio),
- **Minimal maintenance** (read-only root file system—hardly ever breaks),
- **Automatic recovery** (restarts Kodi if it crashes).

LibreELEC comes with:

- **SSH access** (for remote configuration),
- **Samba (SMB)** network sharing (for uploading videos),
- **Built-in Python support** (through Kodi add-ons).

So instead of managing a full operating system like Ubuntu, LibreELEC focuses purely on **“turning a small board into a media player appliance.”**

---

## What is Kodi?

**Kodi** is the **media-center application** that runs on top of LibreELEC.

It provides:

- A **graphical interface** for playing videos, music, or images,
- Support for **almost every video format**,
- A **plugin system** (add-ons) that lets developers automate and extend its functionality using **Python**.

You can think of Kodi as the “engine” that actually **plays the videos** on the TV screen.

Kodi exposes a **JSON-RPC API** and a **Python API**—allowing external systems (like your Central Hub) to send commands such as:

- *“Play this video file.”*
  - *“Pause playback.”*
  - *“Report which video is playing and how long it has run.”*
- 

## How LibreELEC + Kodi Work Together

When a Raspberry Pi boots LibreELEC:

1. The system immediately launches **Kodi** instead of showing a normal desktop.
2. Kodi automatically loads any **Python add-ons** placed in its `/storage/.kodi/addons/` folder.
3. Those add-ons can do things like:
  - Watch for new videos uploaded from a server,
  - Start playback in full screen,
  - Monitor system health (storage, network, temperature),
  - Send playback status back to a dashboard.

Together they create a **turn-key video-display appliance**.

---

## Example in Project Context

Let's apply this to your **Maruti Suzuki showroom TV system**:

### Setup Example

1. A **Raspberry Pi** running **LibreELEC + Kodi** is connected via **HDMI** to each showroom TV.
2. When powered on, the Pi automatically boots into Kodi's full-screen media interface (no manual input needed).
3. Your custom **Python add-on** (say `plugin.video.showroomplayer`) automatically starts when Kodi loads.

### How It Works in Action

- The add-on connects to the **Central Hub** via the internet.
- The Hub tells the add-on:

```
"Play SR103_TV12.mp4 and report your status every 30 seconds."
```

- The add-on downloads the video, saves it locally in `/storage/videos/`, and tells Kodi to play it.
- While the video plays, the add-on periodically sends back telemetry such as:

```
{  
  "tv_id": "TV12",  
  "current_video": "SR103_TV12.mp4",  
  "position_sec": 128,  
  "storage_free_gb": 6.4,  
  "is_playing": true  
}
```

- If the showroom's internet disconnects, playback continues seamlessly using locally cached videos.
- When the connection is restored, the device automatically syncs and reports again.

### Outcome

- The **Central Hub dashboard** shows:

- Each TV's *online/offline* status,
  - Which video it's playing,
  - How much storage is left,
  - Any playback or download errors.
- The **Showroom staff** never need to touch the TV or Pi; everything happens automatically.
- 

### Analogy (Simple Explanation for Management)

If the TV is just a screen,

then the **Raspberry Pi + LibreELEC + Kodi** setup acts as a **smart USB stick that knows what to play and when**, and also **reports back how it's performing**.