

DHRUV INSTALLATION GUIDE

*What you need. Why it matters.
How to install it. How to prove it works.*

A practical guide for installers, customer IT teams, Dhruv administrators and authorised deployment engineers.

THE INSTALLATION JOURNEY Prepare → Connect → Install → Enrol → Publish → Test → Handover

SoC-first. One verified pilot before a wider rollout.

Start with an approved display and the customer’s existing Dhruv service. Use the server chapters only when installing or upgrading a self-hosted environment. Add live streaming and manufacturer controls only when the selected hardware and release support them.

Document control	Record
Reference / edition	DHRUV-INSTALL-001 / v1.0
Prepared	16 September 2026
Companion document	Dhruv How, What & Why Network Setup Guide v1.1
Primary scope	Hosted CMS access, Android/SoC installation, secure enrolment, publishing and commissioning
Additional scope	Conditional LG/Samsung and Amino workflows; Hetzner and local Mac Docker deployment
Evidence basis	Inspected repository source and linked official documentation; not a live acceptance certificate

Repository snapshot: satinder-source/dhruv-platform
9a372468f73656754fdff5f8b83fddd5e6ad8907

RELEASE BOUNDARY No server, network, screen or account was changed to produce this guide. A documented route or successful build is not proof of deployment, package availability, model compatibility or customer acceptance.

Quick installation: one screen first

WHAT Bring one approved screen into the correct customer/site and prove that its content survives a restart.

WHY A working pilot exposes network, package, licence and display problems before they affect the whole site.

Before you begin

Have the approved device package or installation URL, its integrity/signing record, the actual Player service address, authorised CMS access, one test image/video and the completed site network sheet. Do not assume that a CMS login or an APK alone is a complete installation pack. [R2–R4]

- 1. Prepare the screen.** Record model, serial, firmware, MAC address, location, resolution and orientation. Confirm power, mounting and network readiness.
- 2. Connect.** Use the final site network. Check address, DNS, time and HTTPS before installing software; see page 6.
- 3. Install the approved Player.** Follow the exact platform chapter. Preserve an existing installation unless an authorised replacement/upgrade plan says otherwise.
- 4. Enrol securely.** Use the supported protected provisioning method for this release. Bind one physical device to one logical device and the correct site; see page 9.
- 5. Name and assign.** Use a meaningful room/screen name, the correct capability profile and the required product entitlement. Confirm orientation before selecting content.
- 6. Publish a pilot.** Select approved content → select the exact compatible screen → Publish Now. Confirm visible playback, not just an accepted request.
- 7. Test and close.** Reboot, test the supported offline behaviour, check recovery, remove temporary engineering access and record the result.

PROVE IT Record the physical screen, Player version, package digest, device ID, site, content revision, publication outcome and a timestamped photo or observation. Sign off only the tests actually passed.

Stop only the affected step when a prerequisite is missing. For example, an unapproved live-stream decoder should not prevent an otherwise approved still-image signage pilot. Do not enable unsupported controls to bypass a commissioning failure.

Choose your route and responsibilities

Your installation	Read first	Then
Add screens to an existing Dhruv service	Pages 4–9: release, site, network, CMS and Android/enrolment	Platform chapter, then pages 14–16 and 25–29
Install LG / Samsung / Amino equipment	Pages 4–7, then pages 11–13	Validate the exact model and release before commissioning
Install or update the self-hosted service	Pages 18–24: deployment engineer route	Pilot enrolment and full acceptance before cutover
Use a local Mac Docker test environment	Pages 18, 21–24	Test-only identities/data; do not redirect production screens
Use ChromeOS, Raspberry Pi or Windows	Page 17: separate endpoint approval	Do not infer support from Android instructions

Reader’s map

- 04–06 Release pack, site survey and network checks
- 07–13 CMS setup, Android/SOLUM, enrolment, LG, Samsung and Amino
- 14–17 First publication, schedules, live streaming and other endpoints
- 18–24 Server architecture, Hetzner, database, Docker, build and cutover
- 25–29 Acceptance tests, recovery, troubleshooting and handover
- 30–31 Source register and official installation references

Who does what

The customer IT owner approves addressing, firewall rules, certificates and network access. The installer connects and configures the physical equipment. An authorised Dhruv administrator assigns the customer/site, licences and enrolment. The deployment engineer owns server changes, database migrations and rollback. A named customer representative signs acceptance.

HOW TO READ THE STATUS Repository-confirmed means a source contract was inspected. Conditional means a package, model or deployment approval is still required. Example means replace the value before use. “Passed” requires site evidence; USER VERIFIED requires the user’s own acceptance.

CMS = content management system. Player = software that retrieves and displays assigned content. SoC = the display’s built-in computer. Enrolment = securely binding a device to Dhruv. Commissioning = proving the installed system works. A feature being licensed does not prove that a particular screen supports it.

Obtain and verify the release pack

WHAT An approved, platform-specific release pack supplied by the Dhruv release owner, not an arbitrary file copied from an old chat or build job.

WHY The package, server contracts, signing identity and enrolment method must agree. A source repository is not a public installer catalogue. [R2–R5]

Required item	Installer check
Release identity	Version, immutable source revision, release channel and approval reference.
Platform/package	Correct model/firmware/CPU target, file name, exact package ID and install method.
Integrity and signing	SHA-256 digest plus approved signer identity or platform verification record.
Connection/provisioning	CMS origin, Player origin, approved enrolment delivery method and owner.
Upgrade/recovery	Previous approved version, data-preservation steps and supported recovery process.
Evidence	Release notes, required capabilities, test results and known limitations.

Choose the correct package family

Platform	Expected form	Approval boundary
Android / SOLUM Android SoC	Approved signed .apk	Inspected package ID: uk.org.dhruv.player; not the KDS app. [R2]
LG commercial webOS Signage	Approved .ipk / SI-server deployment	Model-specific installation handoff; package service not verified here. [R5]
Samsung commercial Tizen / SSSP	Signed .wgt / supported custom-app deployment	Model/generation-specific; not an Android APK. [R5]
Amino H200	Vendor-approved firmware/provisioning plus service configuration	Do not substitute a generic Android APK or invent firmware credentials. [R6]

HOW Obtain files through the approved protected channel. Compare the digest with a separately trusted release record before installation. Record results; do not publish secrets or signed download tokens in a handover.

PROVE IT The delivered package, hash, signer, platform and authorised channel all match. No universal public installer URL or “latest approved version” is asserted by this guide.

Survey and prepare the installation site

WHAT A safe physical installation, recorded equipment identity and an agreed operational configuration.

WHY A correctly installed Player cannot fix unsuitable mounting, an unavailable input, a restricted network, an incorrect clock or insufficient local storage.

- 1. Confirm the work area.** Use the display manufacturer's mounting, ventilation, power and handling requirements. Physical mounting and electrical work belong to competent authorised personnel; this software guide is not a mounting specification.
- 2. Identify the equipment.** Record the display and any external decoder separately. Photograph the asset label only where permitted. Confirm the model/firmware combination is in the site's approved support matrix.
- 3. Preserve the existing setup.** Record current app/version, CMS device ID, screen name, schedule and network settings. Do not factory-reset, clear app data or uninstall a working Player just to start again.
- 4. Prepare the connection.** Check the correct HDMI/input where used, Ethernet patch lead, switch port, VLAN and link. For Wi-Fi, obtain the approved unattended connection method; captive portals are unsuitable for this workflow.
- 5. Set the display configuration.** Set the intended resolution/orientation, date/time/timezone, power-on input/app and screen-protection behaviour. Keep manufacturer protection measures unless a reviewed signage setting replaces them.
- 6. Check storage and access.** Reserve capacity for the assigned content and its validated replacement, plus working space. Confirm an authorised recovery route before enabling kiosk lockdown.

Site record	Complete before enrolment
Customer / site / room / screen name	
Display model / serial / firmware	
Player / decoder model and version	
MAC / IP / VLAN / switch port	
Native resolution / orientation	
Installer / IT owner / change reference	

PROVE IT The equipment, location and logical screen record agree. Power, networking, display output and authorised recovery access work before any content is published.

Network readiness before software installation

WHAT Approved addressing, name resolution, time and connectivity from the actual device network.

WHY The CMS, Player, media source and optional management service may use different destinations. A working laptop browser does not prove the screen's network path. [R2, R4; N1]

Flow	Baseline rule / condition
Player → approved Dhruv/media hosts	Outbound TCP 443 for the inspected HTTPS design; use exact approved hosts.
Device → site DNS service	UDP/TCP 53 where conventional DNS is used; use site-approved resolvers.
Device → approved time source	UDP 123 where NTP is used; verify the actual device time mechanism.
Address service	DHCP or an approved static assignment/reservation; follow site relay/NAC policy.
Inbound internet → screens	No unsolicited port-forward required for ordinary Player-initiated signage.
Streaming / local control / support	Additional rules only for the selected protocol, model and support service.

Starting hostnames from the inspected source are cms.dhruv.org.uk for CMS administration and player.dhruv.org.uk for the Android Player. Their mapping to the running approved release must be confirmed; do not substitute a guessed api/core/install hostname. [R2–R4]

Read-only installer checks

```
nslookup cms.dhruv.org.uk
nslookup player.dhruv.org.uk
curl --connect-timeout 5 --max-time 15 -sS -D - \
-o /dev/null https://cms.dhruv.org.uk/api/auth/me
```

Run from an authorised machine on the intended network, then check the real device. An unauthenticated protected API should reject the request; a 401 is not a transport failure. HTML from an API route, a certificate error or an unexpected login redirect needs investigation. Do not use curl -k as proof of correct TLS. [R4, R8]

PROVE IT Check DNS, certificate trust, correct clock and the exact authenticated Player flow. Refer to Network Setup Guide v1.1 for detailed ports, multicast, TLS inspection and endpoint records. Do not open every port listed there. [N1]

Connect to the CMS and prepare the customer

WHAT A signed-in, authorised operator and the correct customer/site configuration before a physical screen is claimed.

WHY Enrolment, publication and device control are separate permissions. Network access does not grant a licence or permission to manage another site. [R3, R4]

- 1. Open the supplied CMS address.** Use the HTTPS address from the release/site sheet. Confirm the expected tenant and sign-in page. Do not accept a certificate warning or use an unknown shared administrator login.
- 2. Sign in using the configured method.** Use the approved Dhruv account or SSO flow. Microsoft credentials belong only on the Microsoft sign-in page, not in a Dhruv device field. [R3]
- 3. Select the right scope.** Confirm customer/account, site, timezone and the exact target screen group. Reuse valid existing records instead of making duplicate customers or devices.
- 4. Check licensing and capacity.** Confirm the required product and feature entitlement and any device allocation. Do not assume that one product licence unlocks another product, or that licensing implies hardware compatibility.
- 5. Prepare folders and access.** Create or select the permitted Content Library folder. Give daily operators only the access required for their role; keep credential management with the authorised administrator.
- 6. Test the enrolment administration route.** The maintained path is `/cms/admin/configurations/device-credentials/`. Displayed navigation labels can vary with the deployed shell. Missing access must be resolved as a permission or release issue, not bypassed. [R3]

Who can issue enrolment credentials?

The inspected contract requires a verified human `super_user` or `platform_admin` with `device-credentials:manage`. A tenant administrator alone is excluded. Sensitive credential operations require recent step-up confirmation; the documented window is five minutes. [R3]

PROVE IT The installer can see only the correct site and compatible screens. An authorised administrator can perform the required step-up. A normal publishing operator cannot issue or revoke device credentials.

HOSTED INSTALLATION Adding screens to an existing service does not require installing Node.js, Docker, PostgreSQL or a new CMS on the operator's computer. The server chapters are for authorised self-hosting work only.

Install the Android signage Player

WHAT The approved signed Dhruv signage APK on a supported Android device or Android-based commercial display.

WHY The signage Player is distinct from the KDS application. The inspected Android identity is uk.org.dhruv.player; its source minimum is Android API 24, but that is not a complete model/firmware compatibility guarantee. [R2]

1. Verify the release. Check the APK digest and approved signing-certificate record. An APK that is correctly signed by an unknown key is not an authorised Dhruv release.

```
# macOS; use sha256sum on Linux
shasum -a 256 "/approved/path/DhruvPlayer.apk"
# Android SDK Build Tools must be installed
apksigner verify --verbose --print-certs \
"/approved/path/DhruvPlayer.apk"
```

2. Use the approved installation channel. Prefer the site's approved device-management or manufacturer provisioning process. Permit app installation only through that managed policy. [R2; S3]

3. Use USB ADB only when authorised. Install Android Platform Tools on the engineering computer, enable the model-supported USB debugging path and approve that computer on the device. Select the exact serial number. [S1]

```
adb devices -l
adb -s "DEVICE_SERIAL" install -r "/approved/path/DhruvPlayer.apk"
```

The -r option replaces the application while retaining its data. A signer or version incompatibility is a stop-and-review condition: do not resolve it by uninstalling, clearing data, forcing a downgrade or installing a test-signed APK. [S1, S2; R2]

4. Launch and inspect. Start Dhruv using the approved launcher/profile. Confirm the installed version and device identity. "Secure enrolment required" can be the correct unprovisioned state, not an installation failure. [R2]

5. Close temporary access. After provisioning and recovery checks, disable temporary debugging and remove unneeded installation privileges. Do not expose ADB on the internet.

PROVE IT APK integrity and signer match; the correct package launches; its version and logical device identity are recorded. Proceed to enrolment. Fullscreen display alone is not proof of managed kiosk lockdown.

Enrol the device securely

WHAT A one-time, device-bound exchange that gives this Player its own credential.

WHY Sharing credentials or copying a provisioned image between screens breaks identity, audit and revocation. An installation file is not an enrolment credential. [R3]

- 1. Identify the exact device.** Use its existing logical ID or the approved new-device provisioning record. Confirm the hardware binding, site and active entitlement; do not invent a fingerprint or duplicate an existing screen.
- 2. Authorise the operation.** The credential administrator signs in and completes step-up, then creates the single-device enrolment in the maintained credential-management flow. [R3]
- 3. Receive the one-use sidecar.** The documented CMS returns the opaque enrolment as a downloadable .bootstrap file. Its default validity is ten minutes. Do not print, copy, email or paste the raw secret into the guide, a shell command or a screenshot. [R3]
- 4. Deliver through the approved platform bridge.** The inspected Android runtime can consume a managed application restriction named dhruv_enrolment_token. Use the release-approved secure provisioning connector. Do not assume Android can open the Windows .bootstrap association or that copying a file to Downloads provisions it. [R2, R3]
- 5. Allow the exchange.** The device calls POST /api/device-enrolments/exchange, receives its own credential and uses POST /api/device-credentials/verify where required. It must retain the same identity across restart. [R3]
- 6. Confirm and clear.** Check the correct device/site and subsequent authorised heartbeat. Remove temporary sidecars and provisioning material according to the approved delivery process; retain only non-secret evidence.

Expiry, timeout or retry

An expired or consumed enrolment requires a fresh authorised operation. A timeout does not prove a mutation failed: inspect current device state before issuing another secret. Do not use a reusable master token to avoid the one-use flow. [R3]

STOP CONDITION When the approved Android delivery bridge or required hardware-binding process is not supplied, stop credential delivery and ask the release owner to complete it. The package and network checks can still proceed; do not improvise a plaintext-token workaround.

PROVE IT The exact device enrolls once, appears in the right site, rejects inappropriate scope and reconnects with its own protected credential after restart. Raw tokens must not appear in logs or handover evidence.

SOLUM: install without losing existing devices

WHAT The Android installation and enrolment route, with SOLUM-specific firmware and capabilities checked separately.

WHY A SOLUM screen that already has Dhruv installed must not be treated as a blank device. Preserve its identity and known working content while repairing only the missing step. [R2, R3]

- 1. Inventory all screens.** Record display serial, device ID, current Player version, assigned content and last-seen status. Match the physical screen order to the CMS rather than assuming list order is wall order.
- 2. Check the dashboard before enrolling.** For the existing four-screen pilot, retain the agreed prerequisite: confirm that the repaired enrolment-file download is in the deployed dashboard before enrolling screens 2–4. Do not infer deployment from a PR or an available source patch.
- 3. Upgrade only when required.** An existing working installation does not need reinstalling for every content change. For an approved update, use the compatible signed package and preserve its device data.
- 4. Enrol each unprovisioned unit separately.** Use a fresh device-bound enrolment through the approved bridge. Never copy screen 1's credential or a provisioned app-data backup to the other screens.
- 5. Test individually.** Publish a clearly numbered test asset to each physical screen. Check native orientation, full frame, device ID, current assignment and actual playback.
- 6. Retry the intended publication.** After identity and authorisation are correct, inspect pending publications and reissue only the required operation. Keep old content until the new validated assignment is ready.

Multi-screen content is a separate acceptance item

Prove one-screen/one-content operation first. A common playlist or correct screen numbering does not prove a synchronised splash. Record the approved wall profile, slice order, resolutions, readiness and physical timing. The inspected scheduling contract explicitly distinguishes preparation from production activation. [R7]

PROVE IT Keep one result per screen: installed version, enrolment, heartbeat, visible content, reboot recovery and enabled controls. Do not mark the whole wall accepted because one screen works.

The pilot prerequisite above carries forward the existing installation decision; this guide has not inspected those physical screens or confirmed a current dashboard deployment. Windows is not a prerequisite for this SoC installation.

LG commercial webOS Signage

USE ONLY WITH AN APPROVED RELEASE This chapter follows the LG installation handoff. The handoff is not evidence that an install endpoint, .ipk or hardware model is production-approved. Obtain the exact model/firmware instructions and release pack before continuing. [R5]

WHAT A native Dhruv Player installed using the supported commercial signage deployment mechanism, without an external player when that exact model is approved.

WHY Consumer television apps and developer-session installations are not interchangeable with a permanent commercial signage installation. [R5]

- 1. Record and validate.** Confirm LG model, serial, webOS generation, firmware and intended orientation against the approved matrix.
- 2. Prepare the display.** Connect the final network, set clock/timezone and confirm the approved power-on and network-standby settings. Keep a local recovery route available.
- 3. Open the model-specific server settings.** Use the commercial SI Server / Server Setting workflow supplied for that generation. Menu wording varies; do not enter a hidden service menu based on another model's instructions.
- 4. Enter the verified deployment endpoint.** The repository describes <https://player.dhruv.org.uk/install/lg> as a target shape, not a verified download service. Enter only the live-checked URL in the approved release sheet. [R5]
- 5. Install the matched package.** Confirm the returned package/version is approved for this model and set Dhruv as the automatic startup app through the supported mechanism.
- 6. Reboot and enrol.** Follow the actual protected enrolment method supplied with that release. A code/QR claim screen is a handoff target; do not assume it exists or replace canonical security with an invented code.
- 7. Commission.** Test image/video, orientation, reboot, network loss/recovery, update recovery and only the controls reported as supported.

PROVE IT Record the installed package/version and one complete restart-to-playback result on the exact LG model. A web page appearing once is not evidence of offline caching, automatic startup or scheduled power.

LG Developer Mode and debugging tools belong in a separate authorised lab workflow, not the customer's permanent deployment path. When native support is not approved, use a separately approved endpoint design rather than forcing unsupported firmware. [R5]

Samsung commercial Tizen / SSSP

USE ONLY WITH AN APPROVED RELEASE A signed .wgt, compatible SSSP/Tizen generation, verified package endpoint and tested provisioning method are prerequisites. This guide does not certify those items. [R5]

WHAT A native Dhruv custom application on a supported Samsung commercial signage display.

WHY Tizen does not install the Android signage APK. Application playback and Samsung hardware-control transport are separate capabilities. [R5]

- 1. Identify the generation.** Record model, serial, firmware, Tizen/SSSP version and orientation. Consumer televisions are not an assumed supported target.
- 2. Prepare network and display behaviour.** Set the correct clock, network, input/power-on behaviour and manufacturer-recommended screen protection. Keep settings recovery available.
- 3. Use the correct installation menu.** Follow the generation-specific Play Via / Custom App / URL Launcher or supported App Management workflow in the approved device sheet. These names are not a universal menu sequence.
- 4. Use only the approved package service.** The handoff target is <https://player.dhruv.org.uk/install/samsung>. Treat it as unverified until the release owner provides live package and certificate evidence. [R5]
- 5. Install and start automatically.** Install the approved signed application; confirm version and startup selection. Reboot and ensure Dhruv starts without an operator opening the launcher.
- 6. Enrol and publish a pilot.** Use the provided secure enrolment flow for this build, bind the correct site, then test a still image and compliant video before scheduling.

Hardware controls are not assumed

Expose power, volume, source, brightness, reboot or capture only when the actual adapter/model profile confirms them. For the OM55N family, the repository requires physical standby/wake and command-outcome validation; a successful LAN connection does not prove every control. [R12]

PROVE IT Record application startup after reboot, the exact enrolled screen, visible media playback, supported controls and recovery behaviour. A command queued in CMS is not an acknowledged or physically completed command.

Do not distribute an Android RustDesk client to Tizen or represent developer-mode access as permanent customer support. Manufacturer-native support and debugging remain separately authorised paths. [R5]

Amino H200: connect and provision the decoder

WHAT An approved H200 installation for the selected IPTV/live-streaming use case, with separate media and management configuration.

WHY Connecting HDMI and Ethernet is only the physical step. The source, stream format, service access and Dhruv integration must also be approved. The repository's hotel-IPTV/H200 adapter remains an isolated workstream with vendor/device acceptance boundaries. [R6; S4]

1. Prepare the unit. Confirm the H200 model/firmware, approved power supply, display input and location. Follow the supplied Amino quick-start instructions for physical connection. [S4]

2. Connect to the intended network. Use the assigned network settings and check the actual clock/DNS path. Record decoder and display identities separately.

3. Apply the approved provisioning profile. Use the assigned vendor/management service and supported configuration process. Do not invent a default login, firmware image, activation code or Dhruv APK installation method.

4. Separate the two paths. Configure the stream/channel receiver independently from any Amino management connection. Allow only the management endpoints needed for the assigned service and region; use the companion network guide. [N1]

5. Configure the source. Record playback URL or multicast group/port, authentication owner, codec/audio, expected latency and any content/DRM entitlement. A licence to use Dhruv does not grant content-broadcast rights.

6. Bind through the approved integration. Use the provided Dhruv device/room mapping. Do not assume that the independent H200 service is installed inside the main CMS or shares the Android enrolment implementation. [R6]

7. Test the installation. Confirm video and audio, selected channel, correct input, sustained playback, source interruption and restart recovery. Test management connectivity separately.

PROVE IT Record decoder firmware, approved service profile, stream identity, physical display result and recovery. Cloud management being online does not prove playback; playback does not prove remote management.

A stream being configured in CMS does not make the CMS a media relay or transcoder. The selected architecture must state where ingest, packaging and playback occur. See page 16 and Network Setup Guide v1.1. [N1]

Publish the first content and verify the result

WHAT One approved, correctly oriented asset published to one known compatible screen.

WHY This validates the whole installation: permissions, media processing, entitlement, device identity, download, local playback and status reporting. [R4, R7]

Prepare the test assets

Use a clear still image carrying the site/screen name, then a short video with an identifiable audio cue where sound is required. Match the approved screen resolution and orientation. Keep the files within the installed release's validated media limits; do not treat a nominal resolution as a codec guarantee.

- 1. Select content.** Open the authorised Content Library folder. Import the image/video, wait for processing and choose the approved ready asset. Retain its name, revision and digest/identity.
- 2. Select the screen.** Choose the exact enrolled screen in the correct site. Confirm its physical identity, orientation and capability, not just a similar display name.
- 3. Publish Now.** Use the maintained Fast Publish action. Select Append or Replace where the workflow requires it, taking care not to overwrite live customer content without approval. [R4, R7]

Verify outside the three-step operator flow

Check publication outcome and current assignment in CMS, then observe the actual screen. Record the asset/manifest identity and timestamp. A successful HTTP response, a queued request or a generic green status is not proof of visible playback.

Test	Pass condition
Still image	Correct screen, full frame, correct orientation and readable content.
Video / sound	Correct aspect ratio, smooth playback and intended audio state.
Manifest / heartbeat	Installed device reports the expected applied content/manifest.
Update while playing	New content is validated before replacing the previous usable cache.
Wrong orientation / scope	Incompatible or unauthorised target selection is rejected or unavailable.

PROVE IT Retain a timestamped observation or permitted photograph of the real screen and the matching CMS evidence. Do not use a design render or preview to sign off installation.

Startup, schedules, controls and offline behaviour

WHAT Unattended restart, correct schedules, supported controls and safe operation during an interruption.

WHY A display that works only until the next reboot is not ready for handover. Content scheduling, screen power scheduling and kiosk lockdown are distinct tests. [R2, R7; S3]

- 1. Verify automatic startup.** Restart the Player, then reboot the device in an agreed test window. Confirm Dhruv launches and returns to the correct assignment without manual input.
- 2. Verify the content schedule.** Create a short test window in the site's IANA timezone. Check start, end and fallback. Record timezone/DST behaviour separately from UTC clock accuracy. [R7]
- 3. Verify power scheduling only when supported.** Test OFF and wake/ON separately, with a person able to recover the display. Standby may change network availability; do not assume successful OFF implies supported ON.
- 4. Apply managed kiosk policy.** Use the manufacturer/EMM-approved dedicated-device method. Android's managed lock-task behaviour needs the correct ownership/policy state. Do not issue a device-owner command or factory reset on an already managed production device without a migration plan. [R2; S3]
- 5. Test loss of connectivity.** After complete asset download, interrupt only the pilot device's network. Verify the release's documented cached/offline behaviour, then restore connectivity and confirm new assignment reconciliation.
- 6. Test controls individually.** For every enabled action, check command result and physical effect. Leave unsupported actions unavailable; they must not disable the rest of the Screen Controls page.

OFFLINE IS NOT UNIVERSAL Binary media cache, web content, live streams and entitlement expiry have different rules. A live feed cannot be assumed to continue without its network source, and a cached Web URL is not the ordinary offline-media path. [R2, R11]

PROVE IT Record app restart, device reboot, schedule transitions, allowed controls and outage recovery. For each test, distinguish "requested", "acknowledged" and "physically observed".

Add live streaming only after signage passes

WHAT A verified source-to-receiver streaming chain with the exact protocols, ports, authentication and decoder support recorded.

WHY The encoder's ingest protocol may differ from the screen's playback protocol. Opening a port does not prove codec, DRM, audio or firmware compatibility. [N1]

Chosen transport	Installation-specific value to supply
HLS / DASH over HTTPS	Playback manifest URL plus every approved segment/key/licence host; usually TCP 443.
RTSP / RTP	Actual RTSP control port; selected TCP interleaving or explicit UDP media configuration.
SRT	Exact UDP port, caller/listener roles, stream ID, encryption policy and latency setting.
LAN multicast	Source address, allocated group, UDP port, VLAN path, IGMP/querier and routing design.
WebRTC	Signalling, STUN/TURN/SFU destinations, selected ports and configured relay range.

This table is a selection worksheet, not a firewall-opening instruction. Detailed protocol guidance and primary references are in the companion network guide. [N1]

- 1. Verify the source.** Confirm a valid, authorised stream on a known compatible reference receiver; record settings without exposing access tokens.
- 2. Verify network capacity.** Size for simultaneous receivers, not just one test screen. As an illustrative calculation, 10 unicast receivers at 8 Mb/s with 30% planning headroom require 104 Mb/s of downstream capacity before other site traffic.
- 3. Configure one supported receiver.** Use the approved Dhruv/decoder integration, not an ordinary file-upload path or an assumed generic browser capability.
- 4. Run the agreed soak test.** Observe picture, sound, buffering, measured end-to-end latency and link/server counters. A proposed pilot duration is 60 minutes, followed by an agreed production-load window; record these as site acceptance criteria, not a vendor guarantee.
- 5. Interrupt and recover.** Test source loss, network loss and restart. Confirm the approved fallback, operator-visible fault state and recovery without manual credentials.

PROVE IT The actual receiver plays the approved stream on the final network at agreed concurrency and recovers correctly. Unsupported live features remain off without blocking ordinary signage.

Other endpoints and optional integrations

WHAT A separate approval sheet for any endpoint or integration outside the primary SoC installation path.

WHY A web page opening on a device does not prove a secure, persistent, licensed and remotely manageable Dhruv Player.

Endpoint / module	Installation boundary
ChromeOS / ChromeOS Flex	Require the approved managed-kiosk approach, device policy, installed runtime and endpoint mapping. Prove persistent enrolment, cache and multi-display behaviour on that hardware.
Raspberry Pi	Require a reviewed OS/runtime package, architecture match, startup service, secure credential custody and recovery procedure. Do not assume a Chromium kiosk command reproduces the Android Player.
Windows	Treat as a separate in-house/prototype route for this guide, not a prerequisite for SoC. Use only its approved package and acceptance record; do not present an engineering MSI as a production installer.
Zeitwart / Qbic / EP700	Use the approved booking/display integration, vendor profile, account and entitlement. Do not apply Android signage or H200 instructions solely because a device is network-connected.
ESL, KDS, EPOS, CRM and AI services	Install only the selected module's reviewed package/connector. These are not automatically part of a standard signage installation.
Remote support	Obtain explicit support approval, named access owner and least-privilege network rules. Do not expose debugging as an alternative to supported device management.

Minimum endpoint acceptance sheet

Record model/OS, package/version/hash, installation source, startup behaviour, supported enrolment path, local secret storage, CMS contracts, offline rules, update/recovery method and tested capabilities. Mark an unverified capability unavailable rather than inheriting it from another platform.

PROVE IT A platform is approved only for the functionality demonstrated on that exact installation. Four connected outputs are not evidence of four independently assigned, synchronised Dhruv screens.

These are approval requirements, not claims that the named optional packages or integrations are presently deployable. The main installation remains focused on the user's SoC-first direction. Retain separate existing workstream ownership rather than creating a second installation architecture.

Self-hosting: release and architecture gate

WHAT The reviewed CMS/Core release, its durable data services, storage, HTTPS gateway and deployment-specific runbook.

WHY The current repository contains multiple components and isolated workstreams. Installing a backend, a frontend preview or the H200 demo stack does not install the complete maintained CMS. [R4, R6, R8]

Layer	Required deployment decision
Public edge	Exact DNS names, certificate strategy, Caddy/Nginx identity and approved route map.
CMS frontend	Maintained build/artefact, actual hosting path or container and immutable revision.
Core API	Reviewed endpoint, actual listener/upstream and durable production composition.
Database	Supported PostgreSQL version, least-privilege roles, migration ledger and backup/restore.
Content / media	Persistent governed content store, media tools and delivery configuration.
Identity / licensing	Durable sessions, audit, device credentials, entitlements and central authority connection.
Optional services	Only modules selected for this site, isolated from core availability.

The deployment owner must supply

An immutable approved image/build reference; the complete Compose or service manifest and override order; service names; private listeners; secret-delivery method; storage mounts; database migration and rollback commands; readiness probes; and authorised cutover steps. No generic manifest in this guide substitutes for that pack.

Sizing is a workload decision

Record CPU architecture, RAM, CPU headroom, free space and expected simultaneous clients. Budget disk for media masters, delivery files, processing workspace, database/log retention, previous images and recovery headroom. Do not infer sufficient capacity from an “80 GB” server label or a small number of operator connections.

PROVE IT Every layer maps to an approved artefact and a named owner. Staging is isolated. The current live configuration has been inventoried and a restore has been rehearsed before a production change.

Existing Hetzner server: discover and preserve

WHAT Read-only inventory of the actual server, edge, CMS and API before any installation or upgrade.

WHY The repository records a historical distinction between the original headend and newer maintained components. Neither an old handoff nor a Dockerfile proves the current live hosting path. [R8–R10]

1. Open the correct server. Use the authorised Hetzner project/server and approved administrator access. Record server identity, environment and change window. Keep a recovery console or alternate authorised connection available.

2. Run read-only checks. Use a reviewed copy of the maintained discovery script from the approved repository checkout. Inspect output privately before sharing it. [R9]

```
# From the authorised repository checkout on the server
bash scripts/cms-headend-discovery.sh
# Supplementary read-only checks
docker compose ls
docker ps --format "table {{.Names}}\t{{.Image}}\t{{.Ports}}"
df -h
free -h
ss -ltn
```

3. Map the real service. Identify the loaded Caddy/Nginx configuration, frontend root or container, Core upstream, active Compose files, persistent mounts, database and exact image/build identifiers.

4. Back up before changing. Create a recoverable database-and-content backup with a consistent cutoff, plus the existing frontend artefact, deployment manifests and edge configuration. Record the restoration procedure and secret-manager references, not secret values.

5. Choose additive staging. Reserve a separate origin, project name, database, volumes and private listener. Do not overwrite production files or repoint live devices while establishing the test environment.

DO NOT RUN AS DISCOVERY Do not use `git reset --hard`, `docker compose down -v`, `docker system prune`, an OS reinstall or unreviewed `curl-pipe-shell` commands. Avoid full `docker inspect/config/environment` dumps because they can reveal credentials.

PROVE IT The live CMS route, upstream and data stores are known; backup identifiers and restore evidence exist. Only then proceed to a reviewed deployment change.

Prepare the server, DNS and HTTPS edge

WHAT A supported, maintained host with controlled administrator access and an approved HTTPS entry point.

WHY Container installation is not a substitute for host security, route correctness or certificate trust.

1. Confirm the operating system and architecture. For an existing server, retain its reviewed platform unless an authorised migration is required. On a fresh host, use the approved supported OS and matching architecture; do not infer an OS version from a past note.

2. Install the approved container tooling when needed. Follow Docker's official OS-specific repository installation procedure and record the resulting Engine and Compose versions. Do not replace a working Docker installation merely to follow this guide. [S5]

```
docker version
docker compose version
```

3. Restrict ingress. Expose only approved HTTPS ingress and any deliberately selected certificate/redirect path. Restrict administrative SSH to the approved management route. Keep API, PostgreSQL and administrative consoles private. [S8]

4. Check host and container filtering. Docker-published ports can bypass some host firewall rules; validate actual published listeners and external reachability. Hetzner Cloud Firewall rules do not replace policy on private-network paths. [S5, S8]

5. Configure DNS and certificates. Use the exact approved CMS/Player names and correct A/AAAA records. Configure a complete trusted certificate chain. Caddy commonly uses 80/443 for automatic HTTPS, with alternative challenge arrangements where deliberately configured. [S7]

6. Preserve route and authentication boundaries. Keep browser-session gates separate from device/media/webhook authentication. Validate the actual proxy upstream and port instead of copying an example that may target another service. [R3, R8, R10]

TRUST CAUTION A local CA trusted by the Mac is not automatically trusted by a screen. The inspected Android release uses system trust anchors; installing a user CA may not make that release trust a local test service. Resolve the approved trust path without disabling TLS validation. [R2; S7]

PROVE IT Correct hostname and TLS chain, expected route responses and only intended listeners are reachable over IPv4/IPv6. Record the loaded edge configuration revision and test a new connection after firewall changes.

Database, persistent storage and secrets

- WHAT** The matching database schema, durable stores, persistent media and least-privilege credentials for the approved release.
- WHY** Development memory/file adapters are not evidence of production persistence. An application restart must not lose users, device identity, content references, credentials or audit state. [R3, R4]
- 1. Confirm the migration baseline.** Read the deployed migration ledger and the release’s dependency order. Different workstreams can reserve the same numeric slot; do not sort all SQL filenames and execute them blindly.
- 2. Create the recovery point.** Back up database and content consistently, retain the previous artefact and rehearse restoration. Schedule a write pause where the approved migration requires it.
- 3. Provision the required roles.** Use the reviewed role-provisioning job. Runtime roles must not receive superuser/migration-owner powers. Device enrolment requires its specific least-privilege function-owner arrangement; follow the release contract. [R3, R4]
- 4. Apply migrations using the authorised job.** The repository exposes db:provision-roles, db:migrate and migrate:production scripts. Their existence is not permission to run all three. The release runbook selects the correct entrypoint and credentials for this baseline. [R4, R13]
- 5. Mount durable data.** Use the approved persistent PostgreSQL and governed content paths. Verify ownership/permissions and media-tool availability. Never replace an existing volume with an empty one just to obtain a healthy container.
- 6. Deliver secrets safely.** Use the supported secret-manager or file-based mechanism. Compose secrets mount files; the application must explicitly read them. Do not invent an unsupported _FILE environment variable or assume .env is encrypted. [S9]

Configuration record	Purpose / requirement
NODE_ENV=production	Production composition and fail-closed requirements.
DHRUV_CMS_ORIGINS	Exact approved browser origin list; no wildcard production shortcut.
DHRUV_DATABASE_URL / SSL mode	Secret-managed least-privilege database connection; follow release TLS validation.
Content/blob and media-tool settings	Persistent governed store and the exact tools/settings required by the selected release.

PROVE IT Migration ledger, durable read/write/restart checks, tenant isolation and a restore test pass. Log only non-secret configuration metadata and evidence identifiers. [R3, R4]

Local Mac Docker: isolated staging installation

WHAT A local test installation with its own identity/data and a reviewed Compose deployment pack.

WHY A local rehearsal allows installation and recovery tests without changing the existing Hetzner service or live screens.

1. Prepare Docker Desktop. Use the official installer for the Mac's Intel or Apple-silicon architecture, launch Docker Desktop and verify Engine/Compose. Confirm the organisation's applicable usage terms. [S6]

2. Obtain the complete approved stack. Use the reviewed site/staging manifest, not the unrelated H200 or CRM example simply because it starts. Confirm image architecture and matching CMS/Core revision. [R6, R8]

3. Isolate the environment. Use a dedicated Compose project, database, volumes, secrets and test identities. Explicit external volumes, bind mounts and fixed container names can defeat project-name separation; inspect them.

4. Validate without printing secrets. Set these example variables to the reviewed files and project. The shell guard below checks that the configuration paths were supplied.

```
# Template: set these three non-secret values first
: "${DHRUV_COMPOSE_FILE:?Set the approved Compose file}"
: "${DHRUV_ENV_FILE:?Set the protected environment file}"
: "${DHRUV_COMPOSE_PROJECT:?Set the isolated project name}"
dc() { docker compose -p "$DHRUV_COMPOSE_PROJECT" \
  -f "$DHRUV_COMPOSE_FILE" --env-file "$DHRUV_ENV_FILE" "$@"; }
dc config --quiet
dc config --services
dc config --images
```

The template assumes one complete manifest. Include every reviewed -f override in its documented order when the release uses several files. Do not print dc config without --quiet into shared evidence. [S10]

5. Prepare real-device access. localhost on a screen means that screen, not the Mac. Use an approved LAN-reachable HTTPS hostname and trusted certificate path, with only required listeners exposed. Do not change public production DNS to the Mac.

PROVE IT The local stack has no production credentials or shared volumes. A designated test client reaches the correct service. A stock Android build with a fixed Player origin needs a release-approved test build/routing plan, not an assumed editable URL. [R2]

Build and validate the maintained source

WHAT An isolated build/test of the exact approved repository revision when the deployment process requires source builds.

WHY A design render, partial module or successful backend test is not the maintained CMS build. Do not build on the live server while discovering its current state. [R8, R10]

1. Select the release revision. Use an authorised separate checkout at the reviewed commit. Confirm HEAD and a clean tracked working tree. Do not use a moving branch name as the release identifier.

```
git rev-parse HEAD
git status --short
node --version
npm --version
```

The inspected backend declares Node ≥ 20 < 23 ; the maintained dashboard Dockerfile uses Node 22. Use the release-approved patched Node 22 environment and the lockfiles. These are source constraints, not a recommendation to deploy an unreviewed base image. [R10, R13]

2. Validate the Core API source. Run the documented dependency installation and test command in the isolated build environment. The inspected backend has no separate compile script. [R4, R13]

```
npm ci --prefix backend
npm test --prefix backend
```

3. Validate the full maintained shell. Use dashboard/production-shell, not a standalone render or alternative prototype. [R10]

```
npm ci --prefix dashboard/production-shell
npm run lint --prefix dashboard/production-shell
npm test --prefix dashboard/production-shell
npm run build --prefix dashboard/production-shell
```

4. Record exact outputs. Capture revision, environment, command exit codes, test summaries and artefact digest. Use only the approved publication/signing pipeline for distributable images or device packages.

NOT A DEPLOYMENT COMMAND `npm run dev`, `vite preview` and `npm start` in an unprotected development environment do not establish a customer-ready installation. Passing the build does not apply migrations, issue credentials or update Hetzner.

PROVE IT The reviewed source produces the intended artefacts and its required tests pass. Record failures honestly. This guide supplies commands but does not claim that these builds or tests were executed here.

Start the approved stack and perform cutover

WHAT A controlled start or upgrade of the reviewed services after persistence, migration and recovery prerequisites are complete.

WHY A container in “running” state is not proof of service readiness, correct authentication or live content delivery.

1. Confirm the change gate. Recheck the approved image digest, full manifest/override order, environment, migration evidence, backup and rollback target. Identify all services affected by startup.

2. Pull and start only the approved release. For the manifest-based workflow on page 22, the following commands are examples to run only inside the authorised deployment window. They can create or replace containers. [S11]

```
# Uses the reviewed dc() configuration from page 22
dc pull
# Run the release-specific migration job before application start
dc up -d --wait
dc ps
```

Use the reviewed equivalent when the deployment is not Compose or the installed Compose version differs. --wait does not replace application-specific readiness tests, particularly when no meaningful healthcheck is defined. [S11]

3. Check the actual endpoints. Verify HTTPS, expected content types, sign-in and protected routes on the target origin. The repository distinguishes /api/health liveness from /api/health/ready readiness; confirm those paths exist in the deployed build and gateway. [R4]

4. Run a staging pilot. Test one enrolled device, current manifest, real playback, reboot and network recovery. Verify unauthorised and wrong-scope requests fail.

5. Cut over through the real edge. Use the reviewed Caddy/Nginx/service procedure. Preserve hostname/authentication boundaries and the previous artefact/configuration. Do not replace the working CMS with a static preview.

6. Observe and decide. Monitor errors, latency, storage and device recovery through the agreed window. If the acceptance criteria fail, use the documented rollback rather than loosening security.

PROVE IT Record the live revision/digest and loaded configuration, authenticated readiness, real-device pilot, browser verification and rollback reference. Do not record “deployed” based on a merged PR or a successful pull.

25 / ACCEPTANCE

Commissioning checklist: core installation

Complete on the final network and actual hardware. Enter PASS, FAIL or N/A with a reason; attach evidence IDs rather than secrets. Proposed test quantities/durations are site planning criteria, not guarantees.

ID / test	Pass condition	Result / evidence
C01 Release / inventory	Package, hash/signer, device model, firmware, identity and site match the approved records.	_____
C02 Network / trust	Address, DNS, clock, certificate and selected destinations work from the actual device.	_____
C03 CMS / scope	Correct user, customer/site and role; unauthorised access is rejected.	_____
C04 Enrolment	One device-bound exchange succeeds; no shared/reusable secret or duplicate record.	_____
C05 Heartbeat / assignment	CMS evidence identifies the same physical device and current applied manifest.	_____
C06 Image / video	Correct full-frame image and compliant video; expected orientation/audio.	_____
C07 Publication update	Replacement validates and appears on the intended screen without losing the last usable content prematurely.	_____
C08 App / device restart	App restart and an authorised device reboot return to the intended content without operator action.	_____
C09 Schedule	Agreed start/end/fallback and timezone operate correctly.	_____
C10 Outage recovery	Documented offline/fallback behaviour is truthful; reconnect retrieves the intended current assignment.	_____
C11 Kiosk / service access	Ordinary users cannot reconfigure the Player; authorised recovery remains available.	_____
C12 Handover	Owners, versions, test evidence, open items and rollback/support instructions recorded.	_____

TEST SEPARATELY Engineering checks, live deployment, physical hardware acceptance and customer acceptance are four different records. One must not be inferred from another.

Pilot device ID: _____ Site: _____

Installer / date: _____ Evidence location: _____

Customer representative / acceptance date: _____

26 / ACCEPTANCE

Optional and negative-path acceptance

WHAT Tests for enabled features and failures that could compromise security or unattended operation.

WHY An installation should fail safely and make limitations visible; a successful happy-path publication is not enough.

ID / selected test	Pass condition / evidence required
O01 Live stream	Approved receiver, full media chain, audio, measured latency and agreed soak/concurrency pass.
O02 Hardware controls	Each enabled command has a supported profile, outcome and physical observation. Test wake separately from OFF.
O03 Multi-screen splash	Exact screen order/profile plus fresh readiness and measured physical synchronisation; no partial-start claim.
O04 Web content	Approved origin/capability, working authentication and supported fallback. No arbitrary internal-network URL bypass. [R11]
O05 Emergency function	Only if explicitly in scope and implemented: authorised activation, recovery and a site-approved test plan. Not inferred from ordinary takeover.
S01 Wrong role / site	Insufficient privilege and wrong-site access are rejected without data leakage.
S02 Expired / revoked enrolment	An invalid token fails. Perform destructive/revocation tests on designated test identities only.
S03 Ports / debug access	Private database/API/debug services cannot be reached from unauthorised networks.
S04 Secret handling	No credential appears in CMS screenshots, logs, shell history, bundles or installer evidence.
F01 Optional module fault	Failure in an optional integration does not crash login, ordinary publishing or unrelated controls.
F02 Server recovery	Authorised restart/restore recovers persistent data and does not silently create empty records.

Evidence for an enabled control

Record device/model, installed versions, action, requester, timestamp, command/result identity, reported acknowledgement and actual observed effect. A screenshot verifies picture content only when the capture path is supported and tied to that device; it does not prove a panel woke from standby.

PROVE IT Mark unused features N/A with a reason. Leave failed optional capabilities unavailable and keep the accepted core service usable. Do not label safety-critical or synchronised functions accepted without their own evidence.

Updates, rollback and replacement

WHAT A controlled method to update software or replace a device without losing identity, data or a safe recovery route.

WHY A hurried reinstall can remove cached media, break signing continuity or revive inconsistent credentials. [R2, R3, R4]

Application update

- 1. Prepare.** Record current version, identity, assignment and backup; verify the new package's approved signer, digest and compatibility. Agree the maintenance window.
- 2. Pilot.** Update one non-critical device first. Preserve its data and enrolment through the supported update method. Test content, reboot, controls and recovery before broad rollout.
- 3. Roll out gradually.** Use the approved management channel and monitor per-device outcomes. Do not claim the whole site updated because a package was uploaded.

Rollback

Use the release's supported recovery path. Android/platform version and data compatibility can prevent a simple downgrade; do not force one with engineering flags. For server schema changes, follow the matching database-and-content restoration plan, not just an older application image. [R2, R4]

Restoring credential state requires reconciliation: credentials issued after a restored backup may need re-enrolment, and revocations or security changes made after that point must not be silently lost. Follow the credential recovery contract with the security owner. [R3]

Hardware replacement

Use the explicit same-tenant/same-site replacement workflow. Record the failed device, revoke its access at cutover and give the replacement a distinct identity and fresh credential. Do not copy a provisioned device image or reuse the failed unit's secret. Reapply compatible assignments through the supported CMS workflow. [R3]

Decommissioning

Unassign content/schedules as appropriate, revoke credentials, remove device-management access, reclaim the permitted allocation and follow the approved data-erasure process. Retain audit and asset-disposal evidence under site policy; confirm the old device can no longer access Dhruv.

PROVE IT The updated or replacement device works and the previous device cannot retain unauthorised access. The server, database, media and credential state are consistent with the selected recovery point.

Troubleshooting by installation stage

Symptom	Check first	Safe next action
CMS works; Player is offline	Different service origins, device VLAN, time/TLS, enrolment and licence.	Test the actual device path; inspect redacted status and firewall evidence.
APK will not install	Wrong platform/ABI, package, signer or version policy.	Compare against the approved release; do not uninstall or downgrade to bypass.
Secure enrolment required	Expected unprovisioned state; missing/expired provisioning material.	Use the device-bound flow and approved bridge on page 9.
.bootstrap download fails	CMS release, session/step-up, browser download, proxy response and operation state.	Confirm the repaired dashboard where required; inspect state before retrying.
Device online; content absent	Wrong screen/site, inactive entitlement, failed media processing or pending assignment.	Publish one approved small test asset; verify applied manifest and real output.
403 / scope error	Role, site binding, active entitlement or revoked credential.	Correct authorised records; do not weaken the gate.
API returns HTML / 502	SPA fallback, incorrect edge route/upstream/port or backend readiness.	Inspect the active route map and actual service; do not treat HTML 200 as API success.
Certificate error on screen	Clock, complete chain, old trust store, interception or hostname mismatch.	Repair trust/time/hostname; never disable certificate validation.
Image plays; stream is blank	Separate media host, token, transport, decoder or DRM support.	Test source and selected receiver chain; do not open every streaming port.
Power OFF works; ON fails	Standby network behaviour or unsupported wake method.	Use approved local recovery; disable unsupported remote wake.
Works until reboot	Startup policy, storage/identity persistence, kiosk owner or input selection.	Test approved startup and data path; fullscreen alone is not kiosk management.
Local Mac works; screen cannot connect	localhost, LAN binding/firewall, DNS/trust or fixed production origin.	Use the approved reachable test origin/build without changing production DNS.

ESCALATION PACK Send the issue time, site/device IDs, versions/digests, exact symptom, expected result, affected step and redacted evidence. Never send raw tokens, passwords, full environment dumps or private signing keys.

Installation handover and acceptance record

Field	Complete for this installation
Customer / site / change reference	_____
Installation date / installer	_____
CMS / Player HTTPS origins	_____
Deployed CMS/Core revision or digest	_____
Device IDs / models / firmware / app versions	Attach the completed per-device asset register.
Approved installer source / hash / signer record	_____
Network record / firewall approval	Attach the site sheet from Network Setup Guide v1.1.
Credential / licence / secret-management owners	Record owners and vault references only; no secret values.
Content folders / first publication / schedules	_____
Accepted controls / streaming / optional modules	_____
Backup / restore / rollback evidence	_____
Support contact / support access / review date	_____

Open items remain visible

Item	Impact / restricted feature	Owner / target
_____	_____	_____
_____	_____	_____

Sign-off

Acceptance layer	Name / date / evidence
Network and server deployment	_____
Physical device and content tests	_____
Customer acceptance / USER VERIFIED	_____

FINAL HANDOVER RULE Leave the installer with an operational recovery route and the customer with the correct daily publishing instructions. Remove temporary engineering access. Do not mark unanswered or untested items complete.

Source register: Dhruv implementation

Repository references were inspected at the snapshot below. They describe implementation contracts, source and handoffs—not the live installed state. Private GitHub links require the appropriate repository access. Recheck the approved release before a later installation.

[R1] Repository snapshot

Source identity used for this edition.

[R2] Android Player README and runtime

Package identity, release restrictions, HTTPS, cache and protected credentials. Managed restriction and fixed origin are in MainActivity.java.

[R3] Device credentials and enrolment

One-use sidecars, roles, step-up, exchange, persistence and recovery.

[R4] Core API README

Runtime, content flow, endpoint contracts and production stores.

[R5] LG / Samsung installation handoff

Conditional native package/service targets and model-aware workflows.

[R6] Independent hotel IPTV / Amino H200

Separate service and unverified vendor/device boundaries.

[R7] Playlists and scheduling

Publication, orientation, schedules, power and takeover gates.

[R8] Production CMS headend recovery handoff

Identify the real hosting path; preserve work, backups and fault isolation.

[R9] Read-only headend discovery script

Read-only, secret-avoiding server inventory.

[R10] Maintained CMS shell package / Dockerfile

Build/test scripts; adjacent Dockerfile defines the inspected shell image.

[R11] Governed Web URL content

Exact capability, origin security, live content and fallback boundaries.

[R12] Samsung OM55N MDC handoff

Model-specific control and physical standby/wake acceptance.

[R13] Backend package / script definitions

Node engine range and named build/migration script contracts.

[R2, additional source] Android MainActivity runtime

[R4, additional source] Core runtime and health routes

[R10, additional source] Maintained CMS Dockerfile

Official references and companion guide

Official documentation consulted for this edition. Vendor and tooling procedures can change; use the supported version chosen in the approved release. These references do not certify a Dhruv installation.

[S1] [Android Developers: Android Debug Bridge](#)

Platform Tools, targeted device installation and temporary debugging.

[S2] [Android Developers: apksigner](#)

APK signature verification and signing-certificate inspection.

[S3] [Android Developers: dedicated devices](#)

Managed dedicated-device and kiosk provisioning concepts.

[S4] [Amino: H200 Quick Start Guide](#)

Manufacturer physical setup reference; select the matching hardware guide.

[S5] [Docker: install Engine on Ubuntu](#)

Supported installation procedure and host-firewall cautions.

[S6] [Docker: install Desktop on Mac](#)

Mac architecture-specific desktop installation.

[S7] [Caddy: automatic HTTPS](#)

DNS, ACME challenge paths, certificate persistence and local trust.

[S8] [Hetzner: Cloud Firewalls FAQ](#)

Stateful rules, attachment and private-network limitations.

[S9] [Docker: secrets in Compose](#)

Per-service file mounts; application support for secret-file conventions.

[S10] [Docker: compose config](#)

Manifest validation and service/image enumeration.

[S11] [Docker: compose up / deployment](#)

Container creation/replacement and startup options.

[N1] Companion network guide

Dhruv How, What & Why Network Setup Guide v1.1, supplied with this project. Use its full endpoint, port, streaming, multicast, proxy and firewall-approval material alongside this installation guide. Reconfirm its source snapshot and the live site values before use.

MAINTENANCE Review this guide when the approved installer, enrolment flow, supported model matrix, server topology or public origin changes. Keep the installation guide, network sheet and release pack aligned.