# AGV NFT — Final Deployment Runbook

**Purpose**  
Ship four AGV NFT collections safely to mainnet with correct supply, pricing, phases, and security controls. This runbook is the single source of truth for parameters and the order of operations.

**Contract model (recommended)**  
Deploy **four** separate ERC‑721A “Drop” contracts (one per type). This cleanly enforces different prices/allocations and avoids fragile token‑ID partitioning inside one contract.

Contract A: SeedPass

Contract B: TreePass‑Agri

Contract C: SolarPass

Contract D: ComputePass

If AGV must use a single contract with segmented token IDs, see Appendix C (advanced). Otherwise ignore token‑ID ranges in legacy docs and keep an “ExternalID” attribute in metadata.

## 0) Freeze configuration (do not proceed until all checked)

* Chain: **Polygon mainnet** (or **BNB Chain mainnet**) → choose one and stick to it for all 4 contracts.
* Payment token: **USDT on the chosen chain**. Engineer must paste the **official USDT token address** from the chain’s canonical docs (double‑verify on the chain’s block explorer).
* Royalty standard: **ERC2981** at **5%** to the treasury receiver.
* Primary sale recipient (treasury): SAFE\_TREASURY (a Gnosis Safe address; see step 1).
* Platform fee: **0%** (unless AGV intentionally share fees with a marketplace).
* Owner/admin: **Gnosis Safe only** (no EOAs with admin after step 6).
* Upgradeability: **Proxy (upgradeable)** behind Safe control (or mark **immutable** if AGV prefer).
* Placeholder & reveal: **Optional**; if used, prepare both placeholder and final metadata folders.

**Final supply & pricing parameters** (from AGV's deployment config):

| **Collection** | **Total Supply (on-chain)** | **Public Allocation (claimable)** | **Reserved/Agent (premint)** | **WL Price (USDT)** | **Public Price (USDT)** | **Agent Price (USDT)** |
| --- | --- | --- | --- | --- | --- | --- |
| SeedPass | 400 | 300 | 100 | 29 | 29 | 29 |
| TreePass‑Agri | 600 | 200 | 400 | 55 | 89 | 55 |
| SolarPass | 1500 | 500 | 1000 | 179 | 299 | 179 |
| ComputePass | 299 | 99 | 200 | 499 | 899 | 499 |

**Per‑wallet cap (max mint per wallet)**

* SeedPass: maxPerWallet = 3 (example; choose and freeze)
* TreePass‑Agri: maxPerWallet = 2
* SolarPass: maxPerWallet = 2
* ComputePass: maxPerWallet = 1

If these differ from AGV internal limits, update here now. The contract will enforce them.

**Whitelist windows (UTC)**

* WL start/end per collection (engineer to fill exact timestamps and keep them consistent on the mint page):
  + SeedPass: WL\_START\_SEED → WL\_END\_SEED
  + TreePass‑Agri: WL\_START\_AGRI → WL\_END\_AGRI
  + SolarPass: WL\_START\_SOLAR → WL\_END\_SOLAR
  + ComputePass: WL\_START\_COMP → WL\_END\_COMP
* Public sale starts **immediately after** each WL end.

**Airdrop / Agent premint wallets**

* Provide finalized CSVs per collection: address,quantity (see Appendix A template).

Examples (replace placeholders):

* SeedPass reserved → 0xAIRDROP\_1, 0xAGENT\_1
* TreePass‑Agri reserved → 0xAIRDROP\_2, 0xAGENT\_2
* SolarPass reserved → 0xAIRDROP\_3, 0xAGENT\_3
* ComputePass reserved → 0xNODE\_PARTNER\_1, 0xINVESTOR\_BUNDLE\_1

**Storage**

* IPFS via **thirdweb storage** or **Pinata**.
* Freeze metadata **after** AGV visually QA on testnet; keep final CIDs here:

SeedPass: ipfs://CID\_SEED/

TreePass‑Agri: ipfs://CID\_AGRI/

SolarPass: ipfs://CID\_SOLAR/

ComputePass: ipfs://CID\_COMP/

## 1) Prepare the admin control plane

Create **Gnosis Safe** on the target chain → SAFE\_TREASURY

Signers: 3/5 (or 4/7) with at least **one independent auditor**.

Fund it with native gas + a small USDT buffer for test claims.

(Optional but recommended) Install **Zodiac Delay (Timelock)** module on the Safe

Delay: **24–48h** for role changes / upgrades.

Create a **separate Deployer Safe** (or use a burner EOA only for broadcast): SAFE\_DEPLOYER

If thirdweb requires an EOA for the deploy action, use a fresh hardware‑secured EOA, **then immediately transfer admin to** SAFE\_TREASURY **and revoke EOA roles in step 6**.

## 2) Prepare metadata

1\ Folder structure per collection (example for SolarPass):

/solarpass/

/images/ # 1.png ... N.png

/metadata/

1.json

2.json

...

2\Each metadata/\*.json must include (example):

{

"name": "SolarPass #1",

"description": "Linked to 0.5 kWp PV node with GVT mint rights.",

"image": "ipfs://CID\_SOLAR\_IMAGES/1.png",

"attributes": [

{"trait\_type": "Category", "value": "Solar"},

{"trait\_type": "DAO Weight", "value": "2"},

{"trait\_type": "ExternalID", "value": "S3-0001"} // if AGV keep external numbering

]

}

3、Upload to IPFS. Record the **base folder CID** for each collection.

4、(Optional) Prepare **placeholder** metadata & image if AGV will use delayed reveal.

## 3) Prepare allowlists (snapshots → Merkle)

1. For each collection’s WL, compile a clean CSV of allowlisted addresses:  
   wl\_seed.csv, wl\_agri.csv, wl\_solar.csv, wl\_comp.csv.
2. Generate Merkle roots (AGV can use thirdweb dashboard’s snapshot tool or AGVr in‑house script).
3. Keep: MERKLE\_SEED, MERKLE\_AGRI, MERKLE\_SOLAR, MERKLE\_COMP.
4. Store the raw lists under version control (/snapshots/yyyymmdd/).

## 4) ****TESTNET DRY RUN (MANDATORY)****

Target: Same chain family’s testnet (e.g., Polygon **Amoy** or BSC Testnet).

For each of the four contracts, do the full cycle on testnet:

1. Deploy “**Drop ERC721** (ERC721A, upgradeable)” with:

* Name/Symbol (e.g., SolarPass, SOLAR)
* Primary sale recipient = SAFE\_TREASURY\_TEST
* Royalty 5% to SAFE\_TREASURY\_TEST
* Platform fee 0%
* BaseURI = testnet IPFS CID (or placeholder)

1. Set **Claim Phase 1 (Whitelist)**:

* Start/End = testnet WL window
* Currency = **USDT testnet** address
* Price = WL price from table
* maxClaimablePerWallet = maxPerWallet
* maxClaimableSupply = Public Allocation (this caps claim supply to public number even if total supply is higher)
* Snapshot = corresponding **Merkle root**

1. Set **Claim Phase 2 (Public)**:

* Start = WL end + 1s
* Currency = USDT testnet address
* Price = Public price
* maxClaimablePerWallet = same as above
* maxClaimableSupply = **(no need to set here if Phase 1 already caps to Public Allocation)**
* Note: thirdweb evaluates caps across phases; keep the effective cap = Public Allocation.

1. **Batch airdrop/premint** reserved & agent inventory using the airdrop tool with AGVr CSV.

* Confirm premint **does not** reduce the claimable cap. (It should not; maxClaimableSupply limits claims only.)

1. Test end‑to‑end: WL claim → public claim → royalty on a test secondary → pause/unpause → role transfer to Safe → revoke deployer.
2. Verify source on testnet explorer (or ensure it will verify fine on mainnet).

**Only proceed to production when all four pass.**

## 5) MAINNET DEPLOYMENT (repeat for each of the 4 collections)

**5.1 Deploy contract**

Contract type: **Drop ERC721 (ERC721A, upgradeable)**.

Params:

Name/Symbol:

SeedPass / SEED

TreePass‑Agri / AGRI

SolarPass / SOLAR

ComputePass / COMP

Primary sale recipient: SAFE\_TREASURY

Royalty recipient: SAFE\_TREASURY, fee **500 bps (5%)**

Platform fee: **0%**

BaseURI: ipfs://CID\_xxx/ (placeholder or final)

Record **contract address** and **implementation (proxy) address** for each. Paste them into the deployment sheet and the mint site config.

**5.2 Configure claim conditions (phases)**

For each collection:

**Phase 1 — Whitelist**

Start: WL\_START\_xxx (UTC)

End: WL\_END\_xxx (UTC)

Currency: **USDT mainnet address (chosen chain)**

Price: **WL Price** (table)

maxClaimablePerWallet = maxPerWallet

maxClaimableSupply = Public Allocation

Snapshot: MERKLE\_xxx

**Phase 2 — Public**

Start: WL\_END\_xxx + 1s

Currency: USDT address

Price: **Public Price** (table)

maxClaimablePerWallet = same

(No extra maxClaimableSupply needed if Phase 1 already caps global claimable to Public Allocation.)

**5.3 Batch premint (reserved & agent)**

Run the **airdrop/premint** to deliver the reserved counts to the specified CSV wallets.

Cross‑check totals per collection exactly match the “Reserved/Agent” column.

**5.4 Cap total supply (operational control)**

AGV don’t hard‑set “total supply” in Drop contracts; AGV **control availability** via:

**Premint (reserved)** + **maxClaimableSupply** (public cap).

Ensure AGV do **not** create extra claim phases that lift the cap unless authorized.

**5.5 Security hardening**

Transfer **all admin/minter/pauser/upgrader** roles to SAFE\_TREASURY.

**Revoke** roles from deployer EOA.

If using proxy upgrades: require Safe + Timelock for any upgrade.

Confirm **pause()** is available (or the thirdweb “Disable Claims” switch) and tested.

**5.6 Verify source code**

Verify each proxy + implementation on the chain’s explorer (via thirdweb or manual verify).

Publish the **verified address links** in the mint page.

## 6) Front‑end wiring (mint site)

Hardcode for each collection:

Chain ID

Contract address

USDT token address (the one used in claim phases)

WL/public start times (UTC)

Per‑wallet cap

Audit report URL (if any)

Terms of Sale URL

Show for transparency:

**Contract address** (copyable)

**“View on Explorer”** link

Current **price** (auto‑switch WL/public based on time + allowlist status)

**Countdown** to WL end / public start

**Tx hash** upon success

If AGV support **delayed reveal**, include a reveal banner/timeline and test the reveal flow on testnet first.

## 7) Compliance switches (no surprises)

**Jurisdiction blocking (if required):** enable IP gating/KYC for restricted countries.

**Terms of Sale (checkbox):** user must accept before mint.

**USDT address validation:** render the token’s **official** address in UI tooltips and link to explorer.

## 8) Go‑live sequence (per collection)

**T–24h**

Final WL Merkle uploaded.

Claim phases scheduled (do not “Enable Claim” yet).

Social posts drafted with **exact** contract address & times (UTC).

**T–2h**

Dry‑run one small WL claim on mainnet with a allowlisted internal wallet (if policy allows).

Re‑check Safe owns all roles; deployer EOA has **zero** admin permissions.

**T–10m**

Enable WL phase (if not auto‑starting).

Monitor mempool and contract logs.

**WL → Public**

Public phase auto‑starts right after WL end.

Verify prices switched to public.

Watch claim counts; ensure the **claimable total** never exceeds “Public Allocation”.

**If anomaly** (bot surge, price misconfig, wrong currency):

**Immediate action:** Pause claims (or toggle “Disable Claims”).

Announce temp pause.

Fix config; announce resume time.

## 9) Post‑launch monitoring

Dashboards to watch:

**Claimed vs cap** per collection (expected: ≤ Public Allocation).

**Total supply minted** = Reserved/Agent + Claimed.

**Royalty events** (secondary sales) flowing to SAFE\_TREASURY.

Error rates & failed tx (insufficient USDT approval, allowance issues).

Safe activity (role changes/upgrades should be zero during sale).

Alerts:

X claims/min

Pauses/unpauses

Any admin role change attempt

Drastic USDT price spikes on the wrong token (users selecting fake USDT)

## 10) Handover & freeze

If not planning upgrades, consider **locking upgradeability** (or keep behind Safe+Timelock only).

Publish the final **Deployment Report**:

Chain, contract addresses, verified links

WL/public times, prices, caps

Premint CSV checksums & counts

Role assignments snapshot

IPFS CIDs

## Appendix A — CSV templates (airdrop / premint)

**File name:** seedpass\_reserved.csv (same pattern for other collections)

address,quantity

0xAIRDROP\_1,10

0xAGENT\_1,40

0xMARKETING\_1,50

Sum of quantity must equal the **Reserved/Agent** number for that collection.

## Appendix B — Quick parameter crib sheet

**SeedPass**

WL price: 29 USDT | Public: 29 USDT | Agent: 29 USDT

Public Allocation (claim cap): 300 | Reserved: 100

Max per wallet: 3

Royalty: 5% → SAFE\_TREASURY

**TreePass‑Agri**

WL: 55 | Public: 89 | Agent: 55

Public Allocation: 200 | Reserved: 400

Max per wallet: 2

**SolarPass**

WL: 179 | Public: 299 | Agent: 179

Public Allocation: 500 | Reserved: 1000

Max per wallet: 2

**ComputePass**

WL: 499 | Public: 899 | Agent: 499

Public Allocation: 99 | Reserved: 200

Max per wallet: 1

## Appendix C — (Advanced) Single‑contract with token‑ID partitions

Not recommended for thirdweb Drop, but if mandated:

AGV need a **custom ERC721A** with:

Fixed ranges per “type” (e.g., 001–400 SeedPass, 401–1000 Agri, etc.)

Per‑type pricing, per‑type claim caps, and per‑type WL lists.

Risks:

Complex sale logic & accounting.

Harder audits and higher bug surface.

Front‑end and analytics more complicated.

If AGV go this route, budget time for a full external audit **before** mainnet.

## Appendix D — USDT token address sanity check

**Do not paste random addresses.**

Get the **official USDT address** for AGVr chosen chain from **the chain’s official docs** or **Tether’s official sources**, confirm on the chain’s block explorer, and test with a tiny transfer before enabling claims.

Paste the exact address into:

Each claim phase’s “Currency” field

Front‑end config

## Appendix E — Rollback & emergency actions

**Wrong price / currency / timing:** Pause claims → fix phase → resume.

**Whitelist issue:** Upload corrected snapshot → re‑set Phase 1 with the same cap and extended end time.

**Exploit or critical bug:** Pause claims → announce downtime → if upgradeable, deploy patched implementation via Safe+Timelock → verify → resume.

**Compromised key:** Since admin is the Safe, rotate signers; if a signer is compromised, remove and replace; maintain quorum.

### Final pre‑flight (one‑minute) checklist

* Chain & USDT address verified (official source + explorer)
* 4 contracts deployed & verified (proxy + impl)
* WL & Public phases set with correct UTC times, prices, caps
* Premint CSVs executed, counts match “Reserved/Agent”
* Admin/minter/pauser/upgrader roles → **Safe**, deployer revoked
* Front‑end shows contract addresses, explorer links, Terms of Sale
* Pause tested, Safe timelock active (if applicable)
* Monitoring/alerts live

## ****AGV Protocol – TaskOn Campaign Deployment Guide****

**Objective:**  
Launch a TaskOn campaign to drive NFT minting, community growth, and rGGP token airdrop engagement.

**1. Campaign Overview (Updated)**

**Campaign Name**: AGV Protocol NFT Mint & Airdrop

**Platform**: TaskOn.io

**Chain**: Polygon (low gas, high adoption)

**NFT Contract**: [Paste Thirdweb ERC721 Drop Contract Address after deployment on Aug 13]

**Reward Type**:

**Tier 1**: NFT Minting Reward (SeedPass / TreePass / SolarPass / ComputePass)

**Tier 2**: rGGP token airdrop (delivered via TaskOn reward distribution)

**Duration**: Aug 18 – Sep 8 (21 days total)

**Max Participants**: Unlimited (NFTs capped by supply per tier; airdrop allocation capped by reward pool)

**Special Notes**: Post-campaign hype phase from Sep 9 – Sep 18; main token launch on Sep 19

### ****2. Pre-requisites****

NFT contract deployed and verified via Thirdweb.

Merkle whitelist (if gated minting) generated and uploaded to IPFS.

rGGP ERC20 token contract deployed (if token rewards are included).

Reward pool funded in advance to the reward distribution wallet.

Gnosis Safe or multisig for campaign reward custody.

### ****3. Task Structure****

**Mandatory Tasks:**

**Follow X (Twitter) Account** – AGV Protocol Official

**Retweet Campaign Post** – Containing NFT mint link

**Join Telegram Community** – AGV Official TG group

**Mint NFT on-chain** – Verify via contract call

**Connect Wallet to AGV dApp** – Via Thirdweb login

**Optional Bonus Tasks:**

Invite friends (referral bonus)

Complete governance poll on Snapshot

Stake NFT on AGV platform for extra rewards

### ****4. Chain Interaction Setup****

**For On-chain Tasks in TaskOn:**

Contract Address: [NFT Drop Contract Address]

Method: claim() or mint() (as per Thirdweb config)

Parameters: Auto-filled from Merkle whitelist if applicable

Network: [Select same chain as deployment]

**For Token Claim:**

Contract Address: [rGGP ERC20 Claim Contract]

Method: claimTokens()

Verification: TaskOn checks transaction hash

### ****5. Reward Allocation****

| **Reward Tier** | **Requirement** | **Reward Quantity** | **Distribution Method** |
| --- | --- | --- | --- |
| SeedPass Holder | Mint SeedPass NFT | 50 rGGP | Auto-claim on TaskOn |
| TreePass Holder | Mint TreePass NFT | 100 rGGP | Auto-claim on TaskOn |
| SolarPass Holder | Mint SolarPass NFT | 200 rGGP | Auto-claim on TaskOn |
| ComputePass Holder | Mint ComputePass NFT | 300 rGGP | Auto-claim on TaskOn |

**6. Campaign Flow (Updated for Aug 18 Airdrop & Sep 19 Token Launch)**

**Day 0 – Aug 13**: Deploy NFT contract & token contracts (Thirdweb)

**Day 1 – Aug 14**: Fund reward pool, upload whitelist/IPFS, configure TaskOn campaign

**Day 2 – Aug 15–16**: TaskOn QA testing with internal wallets; fix any integration issues

**Day 3 – Aug 18**: Campaign launch – public announcement across X/TG/Discord, start user onboarding

**Day 7 – Aug 25**: Mid-campaign performance review; adjust marketing spend and boost ads if needed

**Day 14–21 – Sep 1–8**: Campaign end; finalize participant list, trigger automatic reward distribution via TaskOn

**Day 22–26 – Sep 9–13**: Handle support tickets, verify delivery, prepare final report

**Day 27–31 – Sep 14–18**: Pre-launch hype phase – teaser posts, exchange announcements, partner AMAs

**Sep 19**: Main token launch (aligned with campaign momentum)

### ****7. Tips for Maximum Impact****

Use short, high-contrast campaign visuals for TaskOn thumbnail.

Link directly to NFT minting page (avoid multi-step navigation).

Offer referral rewards to boost viral growth.

Sync campaign start with major announcements (e.g., exchange listing, partner news).