📘 REGENEDAG WHITEPAPER — PART II

(Architecture,Tokenomics, Workflows, Fraud Prevention)

Prepared by: Mustapha Ismail Ogunleye (Mioha)

Emails: mustymioha0107@gmail.com

mustaphaismail0107@gmail.com

7. Tokenomics

The REGEN token powers the entire economic and operational ecosystem of RegenDAG.
Its design ensures:

Sustainable distribution.

Transparent aid delivery.

Community rewards.

Long-term protocol growth.

Governance capabilities (Wave 4+).

7.1 REGEN Token Distribution Model

Category	Allocation	Purpose
Relief & Impact Fund	35%	(Direct aid payouts to beneficiaries).
Community & Donor Rewards	25%	(Incentives, staking, engagement rewards).
Governance Treasury	15%	(DAO-controlled strategic decisions).
Development & AI Infrastructure	10%	(Engineering,machine-learning, oracles).
Network Validators	7%	(Incentives for maintaining BlockDAG infrastructure).

Reserve & Sustainability Buffer 5%
(Emergency liquidity for crises).

Education & Inclusion Fund 3%
(Humanitarian literacy programs)

Total Supply (Wave 2): 1,000,000 REGEN

Future versions may enable inflationary or deflationary mechanisms depending on DAO governance.

7.2 Token Utility

Aid distribution

Incentives for responders

Reputation-based bonuses

Future governance voting

AI oracle fee model (Wave 4+)

Donor staking rewards

REGEN is a functional utility token, not an investment instrument.

8. Eligibility & Fraud-Prevention Layer

Humanitarian systems are vulnerable to manipulation. RegenDAG solves this using multi-layer verification.

8.1 Household Mapping

Every registered user is tied to a householdId.
Rules enforced by the smart contract:

1 household = 1 payout.

No double claiming.

No multi-wallet abuse.

No family-level exploitation.
This is the first of its kind in Web3 humanitarian systems.

8.2 AliefVerification Engine

The Alief contract flags or clears users based on simple Boolean logic (Wave 2).

Wave 4 Upgrade Options:

Machine learning analysis

Identity correlation

Geographic verification

Social graph mapping

Oracle inputs from NGOs

Image/video metadata models

This creates a scalable fraud-deterrent across borders.

8.3 Eligibility Checks

Before a payout occurs, all conditions must pass:

✓ Registered(wallet)
✓ Verified(wallet)
✓ Not previously claimed
✓ Household not paid
✓ Alief.isEligible(wallet, householdId) == true

If ANY check fails ʼn payout is rejected.

This guarantees fairness and prevents corruption.

9. Aid Distribution Workflow

This section describes the full lifecycle of a payout.

9.1 Registration Workflow

RegenDAO ’n register(wallet, householdId)

’

AidDistribution stores:
- wallet address
- householdId
- registered = true

Event emitted:

Registered(wallet, householdId)

9.2 Verification Workflow

DAO validates identity:

RegenDAO ′n verifyBeneficiary(wallet, status)

If status = TRUE, the user becomes eligible for final checks.

9.3 Payout Workflow (Critical)

RegenDAO ′n releaseAid(wallet, amount)
′

Check 1: Registered?
Check 2: Verified?
Check 3: Not claimed before?
Check 4: Household not paid?
Check 5: Alief.isEligible == true?

When all conditions pass:

```
RegenToken.transfer(wallet, amount)
```

Event emitted:

```
AidReleased(wallet, amount)
```

This creates a public, immutable audit trail.

10. Smart Contract Architecture (Deep Dive)

Below is the full ASCII architecture diagram that will appear in the PDF.

```
┌──────────────────────────────
────┐
│          RegenDAO          │
│ (Owner / Governance Unit) │
└──────────────┬──────────────
─────┘
│
▼
┌──────────────────────────────
────┐
│      AidDistribution      │
│   - registration          │
│   - verification          │
│    - household control         │
│    - REGEN payouts               │
└──────────┬──────────────┬─────
──────┘
│               │
│               ▼
│     AliefVerification
│     (Fraud / Flag Engine)
│
▼
RegenToken (ERC20)
```

(Mintable by DAO for aid pools)

11. Security & Protection Model

RegenDAG is engineered with best-in-class security design.

✔ Access Control

Only DAO/owner can modify critical state.

✔ Reentrancy Protection

AidDistribution uses ReentrancyGuard to protect payouts.

✔ Immutable Event Logs

Every payout is traceable.

✔ Modular Upgradability

Future upgrades via DAO proposals.

✔ Oracle Isolation (Wave 4)

Prevents corrupted ML inputs from altering eligibility.

12. System Environment

RegenDAG is optimized for BlockDAG but compatible with any EVM chain. Tested on:

BlockDAG IDE

Hardhat local node

Ganache

Remix

The system is designed for multi-network deployment once governance matures.