
DAO
Decentralized Autonomous Organization

WHAT IS IT?

An organization governed by blockchain-based smart contracts that operate autonomously according to encoded rules

PURPOSE

to ensure that all decision-making, funding allocation, and operational enforcement occur through immutable, pre-programmed contracts rather than community voting or administrative oversight.

Executive Summary

Autonomous Smart-Contract DAO to automate digital-health governance through immutable code.

It integrates a **mental-wellness app** and a **secure health-data platform** to manage consent, rewards, and compliance automatically.

Smart contracts ensure **transparency, fairness, and accountability** in every operation.

Digital-health platforms face **governance challenges** in consistency and transparency.

Human oversight can lead to delays or uneven decision-making.

Community voting can be inefficient or unclear.

A **smart-contract governance model** offers automated, unbiased rule enforcement

PROBLEM/NEED

Autonomous Smart-Contract DAO to manage operations through code.

Replace manual or community oversight with **predefined, immutable rules**.

Automate decisions on **consent, compliance, and funding**.

Ensure **transparency, fairness, and accountability** through blockchain execution.

Solution / Concept

Leadership Principals

Ethical Stewardship

rules and limits are encoded to prevent data misuse, enforce informed consent, and preserve user sovereignty

Servant Leadership

automation exists to serve the end-user's welfare through equitable access and error-free execution, not profit maximization

Integrity and Transparency

all operations are traceable on-chain, eliminating hidden decision paths

governance logic aligns data management, privacy, and incentive mechanisms as a unified framework for secure digital-health delivery

- **Systems Thinking**

Proposal & Rule Execution Framework

1. **Initialization:** Founders define governance parameters in the initial contract suite.

2. **Trigger Events:** Smart contracts listen for predefined blockchain or app-level events (e.g., DID authentication, consent grant, completion of wellness tasks).

3. **Validation:** Logic checks confirm authenticity, ownership, and compliance limits.

4. **Execution:** When all conditions are met, the contract self-executes (e.g., issuing tokens or writing compliance proof).

5. **Auditability:** Every action generates an immutable record on-chain viewable by regulators or auditors without exposing PHI.

Objectives of Code Logic

Each function—such as data-access authorization, token issuance, compliance logging, and practitioner verification—is triggered by coded logic tied to verifiable on-chain conditions.

Once deployed, the smart contracts function as the operational charter of the system, guaranteeing transparency, predictability, and auditability across all transactions

DAO Holistic Wellness Mapping

Consent validation	Orthomolecular guidance control	Practitioner verification	Tokenized engagement tracking	SOLUTION
for encrypted health-data sharing	ensuring safe supplement recommendations by cross-checking user medication data	using Verifiable Credentials to confirm authority before data access	rewarding compliance and participation; connects blockchain automation directly to functional health-data workflows	a **trustless and compliant** operating mode

Smart-Contract Summary

AccessControl.sol

governs DID-based identity and data permissions

WellnessToken.sol

mints and distributes HNT token rewards automatically

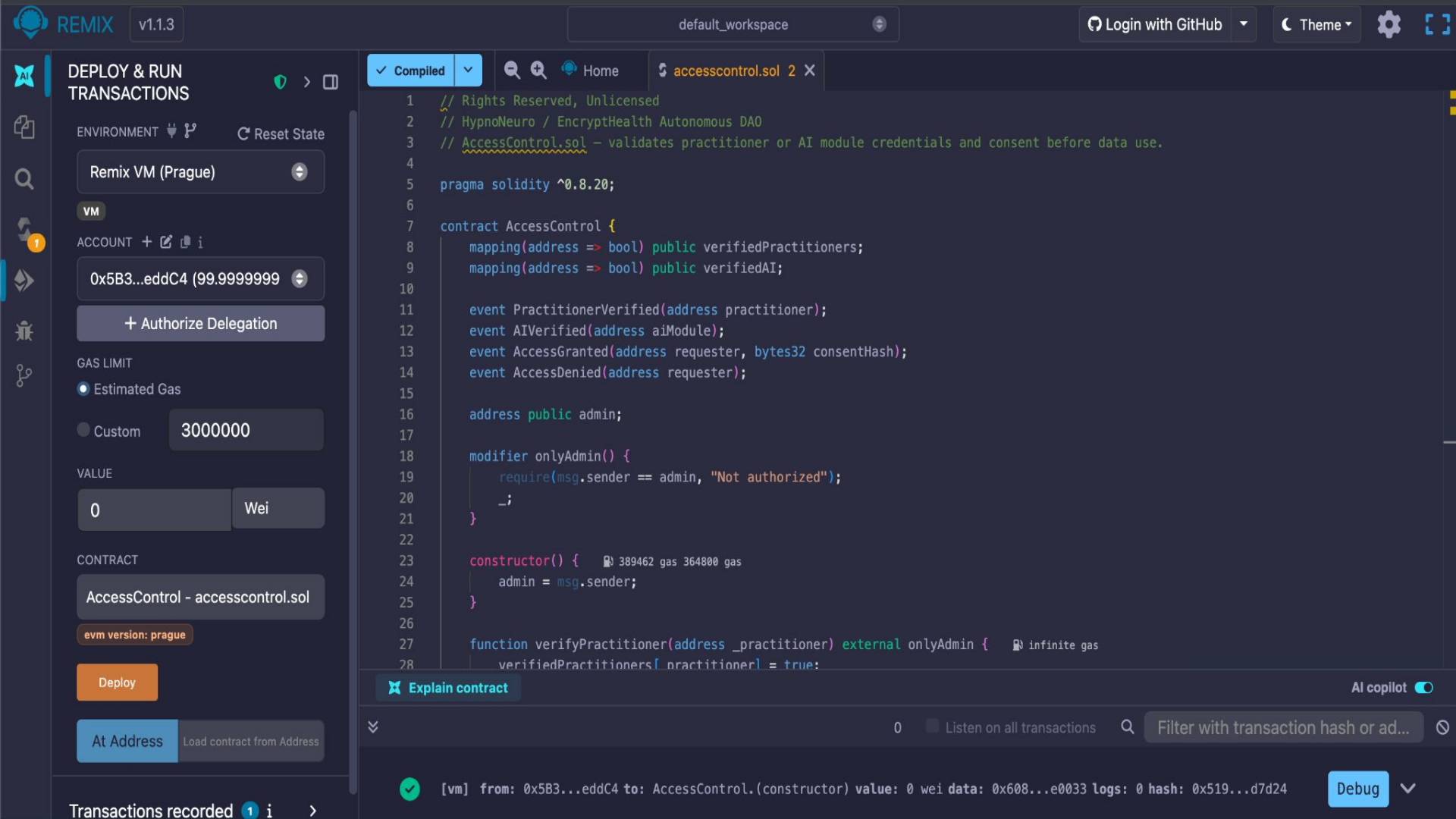
ComplianceLog.sol

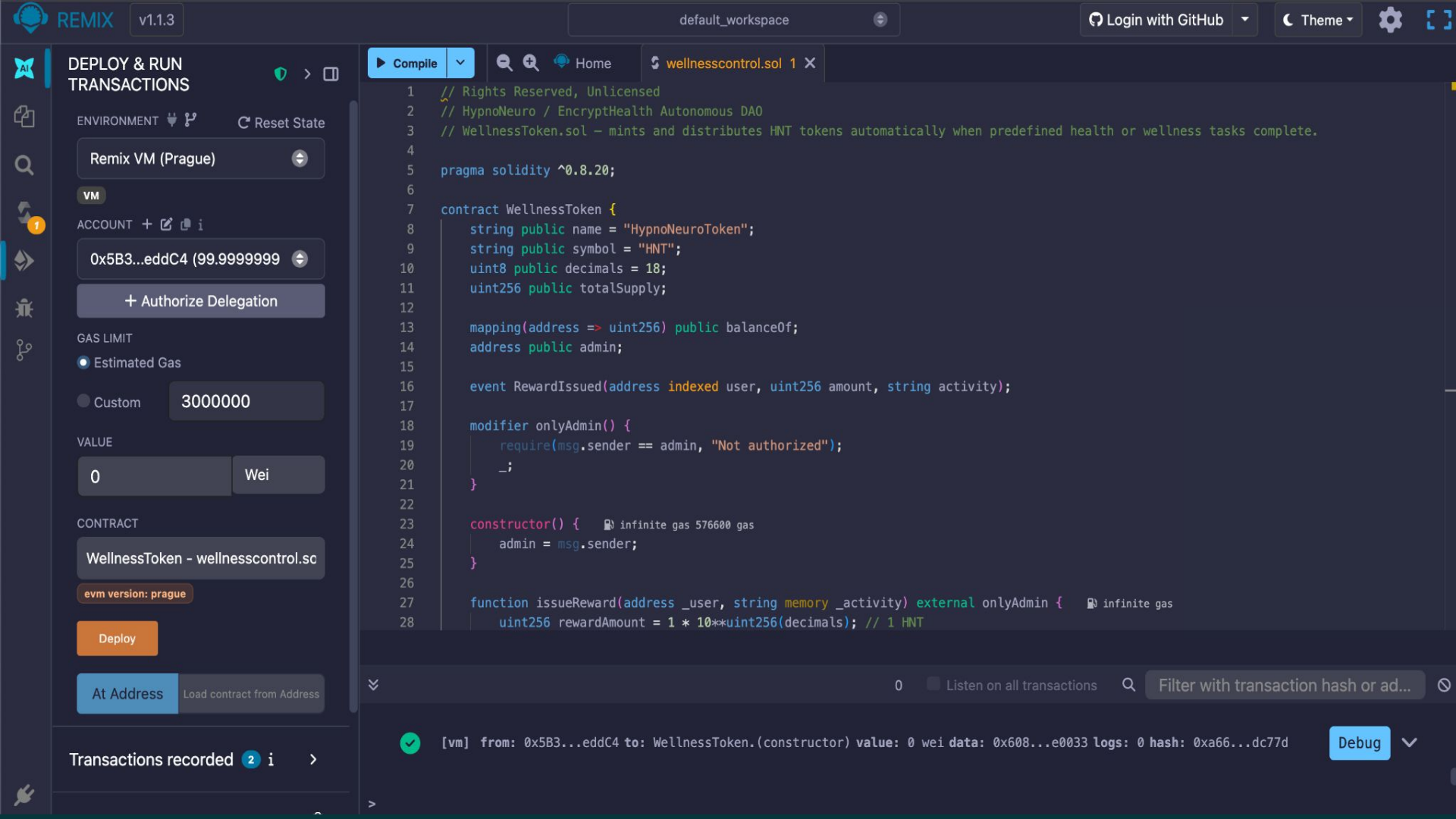
hashes and records data-access events for audit purposes

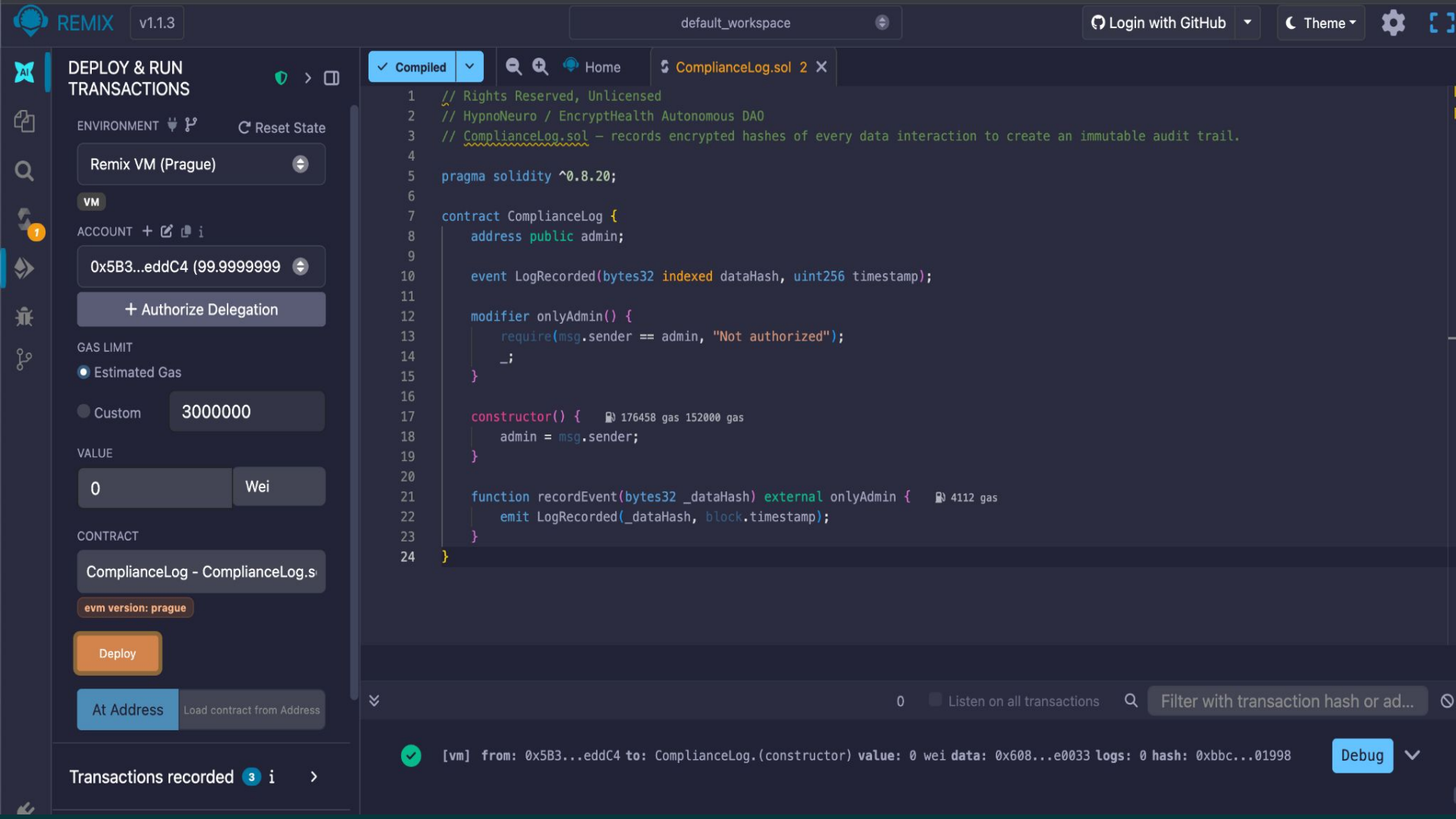
Treasury.sol

executes automatic disbursements based on encoded spending thresholds; verified on a public testnet, enabling independent inspection and legal traceability

Smart Contracts Repository:
github.com/Future-Systems-Lab/autonomous-governance-dao


REMIX
v1.1.3
default_workspace
Login with GitHub
Theme
DEPLOY & RUN TRANSACTIONS
ENVIRONMENT
Reset State
Remix VM (Prague)
VM
ACCOUNT
0x5B3...eddC4 (99.9999999
+ Authorize Delegation
GAS LIMIT
Estimated Gas
Custom
3000000
VALUE
0
Wei
CONTRACT
AccessControl - accesscontrol.sol
evm version: prague
Deploy
At Address
Load contract from Address
Transactions recorded
Compiled
Home
accesscontrol.sol 2
// Rights Reserved, Unlicensed
// HypnoNeuro / EncryptHealth Autonomous DAO
// AccessControl.sol — validates practitioner or AI module credentials and consent before data use.
pragma solidity ^0.8.20;
contract AccessControl {
mapping(address => bool) public verifiedPractitioners;
mapping(address => bool) public verifiedAI;
event PractitionerVerified(address practitioner);
event AIVerified(address aiModule);
event AccessGranted(address requester, bytes32 consentHash);
event AccessDenied(address requester);
address public admin;
modifier onlyAdmin() {
require(msg.sender == admin, "Not authorized");
_;
}
constructor() {
389462 gas 364800 gas
admin = msg.sender;
}
function verifyPractitioner(address _practitioner) external onlyAdmin {
infinite gas
Explain contract
AI copilot
0
Listen on all transactions
Filter with transaction hash or ad...
[vm] from: 0x5B3...eddC4 to: AccessControl.(constructor) value: 0 wei data: 0x608...e0033 logs: 0 hash: 0x519...d7d24
Debug


REMIX
v1.1.3
default_workspace
Login with GitHub
Theme
DEPLOY & RUN TRANSACTIONS
ENVIRONMENT
Reset State
Remix VM (Prague)
VM
ACCOUNT
0x5B3...eddC4 (99.9999999
+ Authorize Delegation
GAS LIMIT
Estimated Gas
Custom
3000000
VALUE
0
Wei
CONTRACT
WellnessToken - wellnesscontrol.sc
evm version: prague
Deploy
At Address
Load contract from Address
Transactions recorded 2
Compile
Home
wellnesscontrol.sol 1
// Rights Reserved, Unlicensed
// HypnoNeuro / EncryptHealth Autonomous DAO
// WellnessToken.sol — mints and distributes HNT tokens automatically when predefined health or wellness tasks complete.
pragma solidity ^0.8.20;
contract WellnessToken {
string public name = "HypnoNeuroToken";
string public symbol = "HNT";
uint8 public decimals = 18;
uint256 public totalSupply;
mapping(address => uint256) public balanceOf;
address public admin;
event RewardIssued(address indexed user, uint256 amount, string activity);
modifier onlyAdmin() {
require(msg.sender == admin, "Not authorized");
_;
}
constructor() {
infinite gas 576600 gas
admin = msg.sender;
}
function issueReward(address _user, string memory _activity) external onlyAdmin {
infinite gas
uint256 rewardAmount = 1 * 10**uint256(decimals); // 1 HNT
0
Listen on all transactions
Filter with transaction hash or ad...
[vm] from: 0x5B3...eddC4 to: WellnessToken.(constructor) value: 0 wei data: 0x608...e0033 logs: 0 hash: 0xa66...dc77d
Debug


REMIX
v1.1.3
default_workspace
Login with GitHub
Theme
DEPLOY & RUN TRANSACTIONS
ENVIRONMENT
Reset State
Remix VM (Prague)
VM
ACCOUNT
0x5B3...eddC4 (99.9999999
+ Authorize Delegation
GAS LIMIT
Estimated Gas
Custom
3000000
VALUE
0
Wei
CONTRACT
ComplianceLog - ComplianceLog.s
evm version: prague
Deploy
At Address
Load contract from Address
Transactions recorded 3
Compiled
Home
ComplianceLog.sol 2
// Rights Reserved, Unlicensed
// HypnoNeuro / EncryptHealth Autonomous DAO
// ComplianceLog.sol - records encrypted hashes of every data interaction to create an immutable audit trail.
pragma solidity ^0.8.20;
contract ComplianceLog {
address public admin;
event LogRecorded(bytes32 indexed dataHash, uint256 timestamp);
modifier onlyAdmin() {
require(msg.sender == admin, "Not authorized");
_;
}
constructor() {
176458 gas 152000 gas
admin = msg.sender;
}
function recordEvent(bytes32 _dataHash) external onlyAdmin {
4112 gas
emit LogRecorded(_dataHash, block.timestamp);
}
}
0
Listen on all transactions
Filter with transaction hash or ad...
[vm] from: 0x5B3...eddC4 to: ComplianceLog.(constructor) value: 0 wei data: 0x608...e0033 logs: 0 hash: 0xbbc...01998
Debug

REMIX v1.1.3

default_workspace

Login with GitHub | Theme

DEPLOY & RUN TRANSACTIONS

ENVIRONMENT | Reset State

Remix VM (Prague)

VM

ACCOUNT

0x5B3...eddC4 (99.9999999

+ Authorize Delegation

GAS LIMIT

Estimated Gas

Custom 3000000

VALUE

0 Wei

CONTRACT

Treasury - Treasury.sol

evm version: prague

Deploy

At Address | Load contract from Address

Transactions recorded 4

Compile | Home | Treasury.sol 2

```
1  // Rights Reserved, Unlicensed
2  // HypnoNeuro / EncryptHealth Autonomous DAO
3  // Treasury.sol - executes automatic, rule-bound disbursements once spending thresholds and multisig approval are met.
4
5  pragma solidity ^0.8.20;
6
7  contract Treasury {
8      address public admin;
9      uint256 public spendingLimit = 100 ether;
10
11     mapping(address => bool) public authorized;
12     uint256 public authCount;
13
14     event AuthorizedAdded(address indexed account);
15     event AuthorizedRemoved(address indexed account);
16     event Disbursed(address indexed to, uint256 amount, string purpose);
17
18     modifier onlyAuthorized() {
19         require(authorized[msg.sender], "Not authorized");
20         _;
21     }
22
23     constructor() {    822530 gas 726800 gas
24         admin = msg.sender;
25         authorized[admin] = true;
```

0 | Listen on all transactions | Filter with transaction hash or ad...

creation of Treasury pending...

[vm] from: 0x5B3...eddC4 to: Treasury.(constructor) value: 0 wei data: 0x608...e0033 logs: 0 hash: 0x298...a84a0 | Debug

>

Ethical & Legal Considerations

- Protects user autonomy through encrypted, self-controlled data
- Removes bias and human error via automated governance
- Ensures transparent audit trails for accountability
- Aligns with HIPAA and GDPR privacy principles
- Embeds ethical leadership directly in smart-contract code

Reflection / Lessons

My understanding of ethical governance led me to design a decentralized system where leadership runs on automation instead of authority. By embedding clear rules into smart contracts, bias and human error are minimized, and decisions stay consistent.

This project showed that strong governance can exist through transparent, reliable systems built on integrity rather than personal control.

AI-Use Disclosure

This presentation was authored by Margarita Davenport and later refined using generative AI (OpenAI ChatGPT) for grammar, structure, and formatting improvements. The following type of prompt was used: *"Polish this section for clarity and academic tone while keeping the original meaning."*

All ideas, analysis, and final content were created, reviewed, and approved by the author in full compliance with the Colorado State University Global Artificial Intelligence Tool Guidelines.

References

Colorado State University Global. (2024). *Artificial intelligence tool guidelines.* https://csuglobal.edu/student-policies/student-rights-policies

Appendix A - Governance Table (Autonomous Smart-Contract DAO)

Function	Purpose	Logic / Verification	Outcome
Identity Validation	Confirm practitioner or AI authenticity.	verifyAccess() checks DID signature.	Access granted or denied.
Consent Management	Verify EncryptHealth user consent.	Compares consent hash in ComplianceLog.	Data access approved or blocked.
Data Logging	Create immutable audit record.	recordEvent() emits hash + timestamp.	Transparent, PHI-free record.
Token Issuance	Reward verified user wellness tasks.	issueReward() mints 1 HNT per activity.	Tokens added to wallet.
Treasury Control	Manage funds under rule-based disbursement.	disburse() requires multisig approval.	Funds released, logged.
Upgrade & Compliance	Ensure security, audit, and regulatory proof.	Audit hash verified via oracle.	Updated, verified system state.