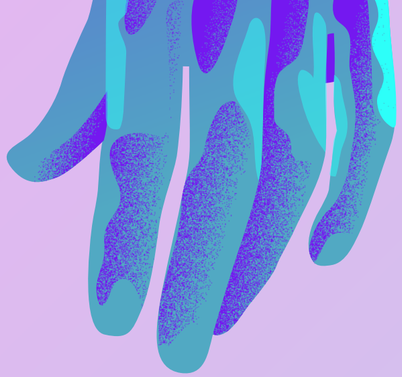


a multi-agent rag system for auditable token distribution

Presented By : Har Sze Hao 22ACB05703

Supervised By: Dr. Aun Yichiet

Faculty of Information Communication and Technology

research background and motivation

- Data inconsistency and delayed services (data silos among government departments)
- Lack of immutable audit trails affecting transparency in fund disbursement tracking
- Citizens' lack of control over personal information
- how ai agents can play a role in government services

news

Middle-class resisting Padu registration due to security and privacy concerns

NATION

Saturday, 17 Feb 2024

PETALING JAYA: There is still strong resistance from the M40 and T20 groups in registering on the Central Database Hub (Padu), with many citing security and privacy concerns, difficulty in registering, or, more worryingly, not seeing why they should bother.

Padu developed using outdated methods, says cybersecurity expert

1 YEAR AGO

Chia Wan Rou

Share

Fong Choong Fook says Padu may be vulnerable due to outdated development methods, raising concerns about data integrity and security.

01.

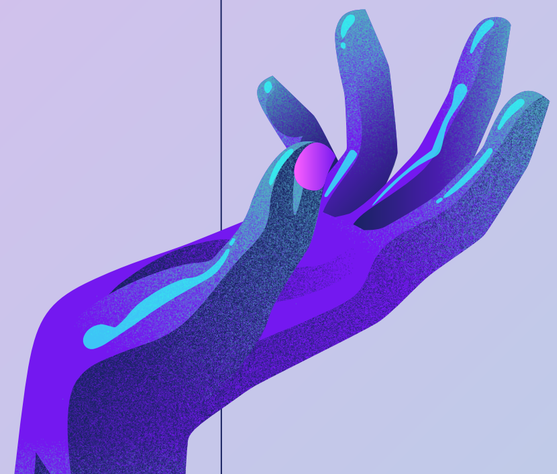
To develop a **quantization method** to synthesize government policy **using ai agents + RAG and LLM** to determine eligibility for subsidies.

02.

To **implement an ERC-20 token,** named MMYRC (Mock Malaysia Ringgit Coin), for airdropping funds to eligible citizens.

03.

To evaluate the application's effectiveness in **improving auditability and traceability of funds** compared to **existing systems**.



objectives of the project

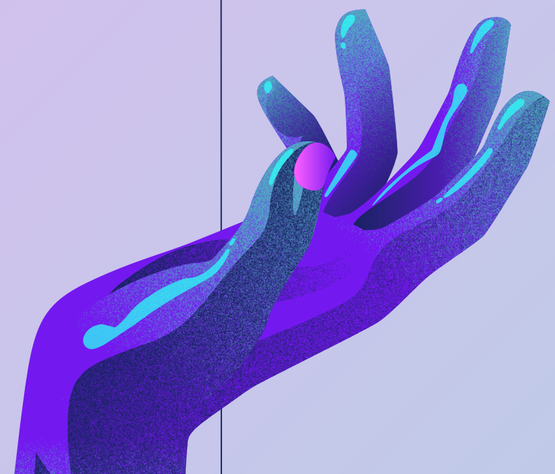
objectives of the project

04.

to explore the use of **zero-knowledge proofs** (ZKPs) in a mock trusted setup with LHDN, allowing citizens to prove their income bracket (e.g., B40-B1, M40-M1, T20-T1) without revealing exact income values, thereby ensuring privacy preservation.

05.

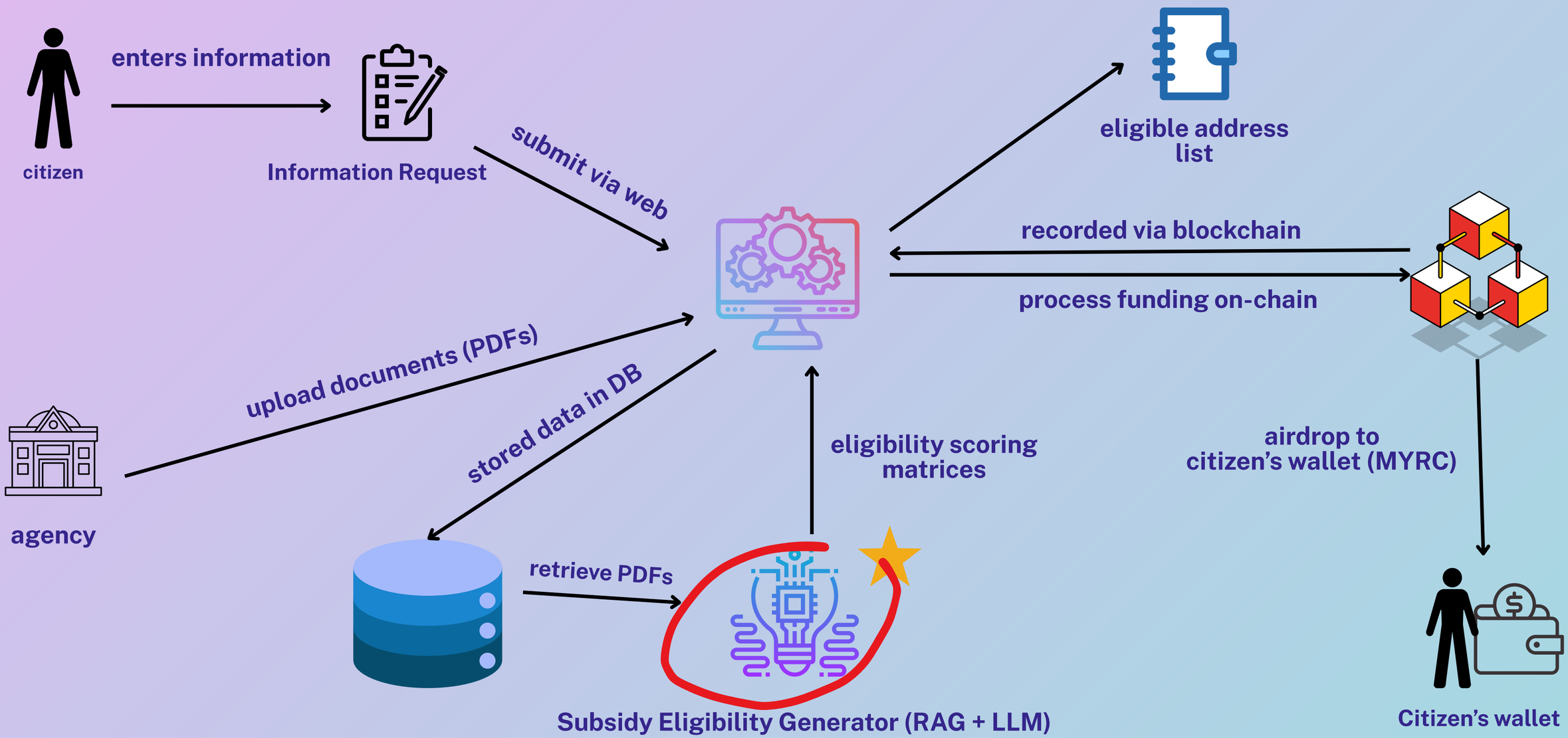
To integrate **smolagents** (a agentic framework) into a dual-analysis framework that combines formula-based burden scoring with RAG-based reasoning, enabling comparison between interpretability and flexibility in eligibility assessments.



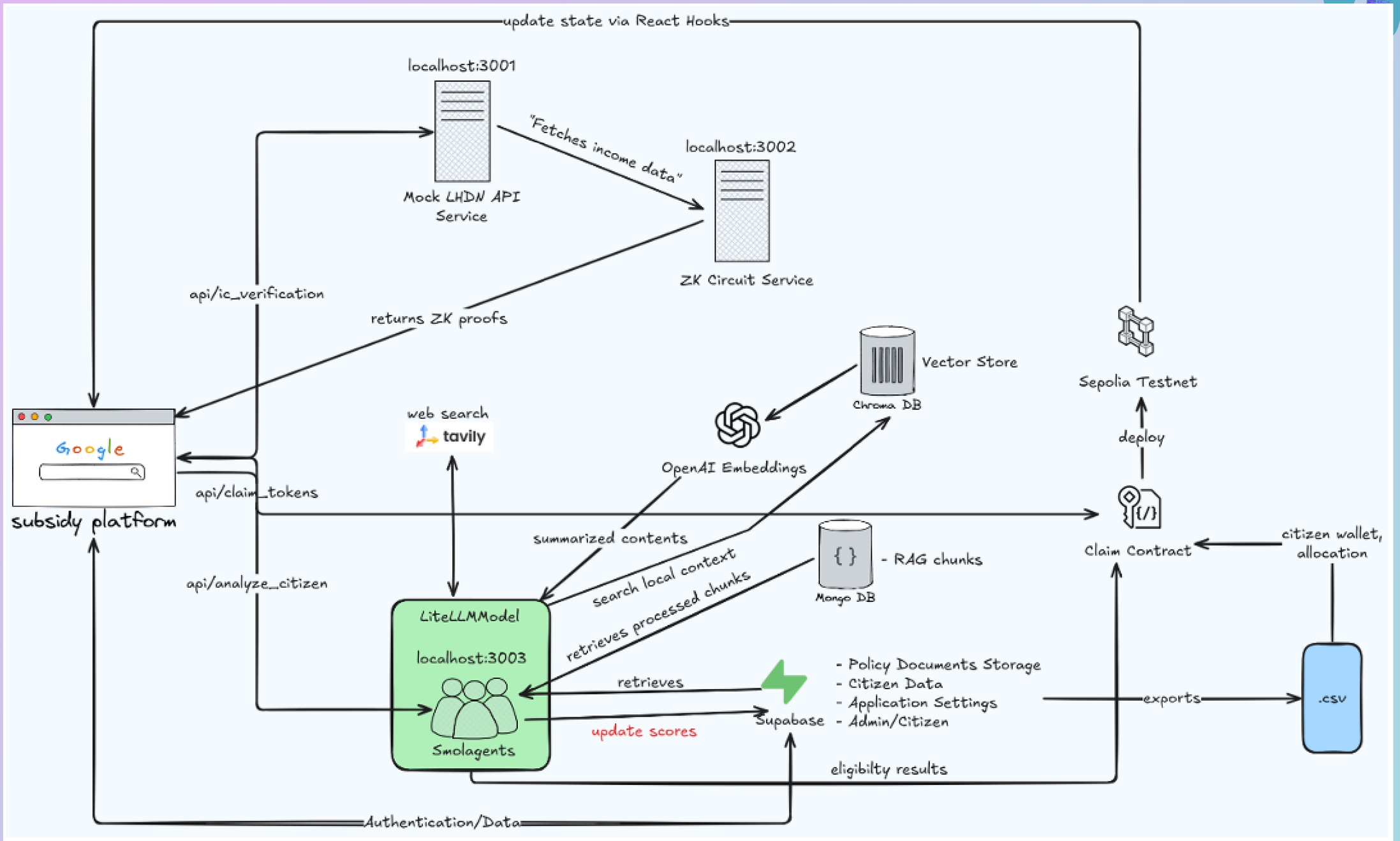
Feature	Traditional System (PADU)	Proposed Blockchain RAG System
Rule Setup	Manual, Static	Data-driven, LLM-generated
Transparency	Limited	Full on-chain tracebility
Fraud Risk	High	Low(smart contract enforced)
Customisation per Region/Group	Hardcoded	Data-contextualized scoring
Real-time Auditability	No	Yes
Document Handling (PDFs)	Manual	Automated via RAG

Table 2.2 Comparison of proposed system over traditional PADU

proposed system architecture (fyp1)



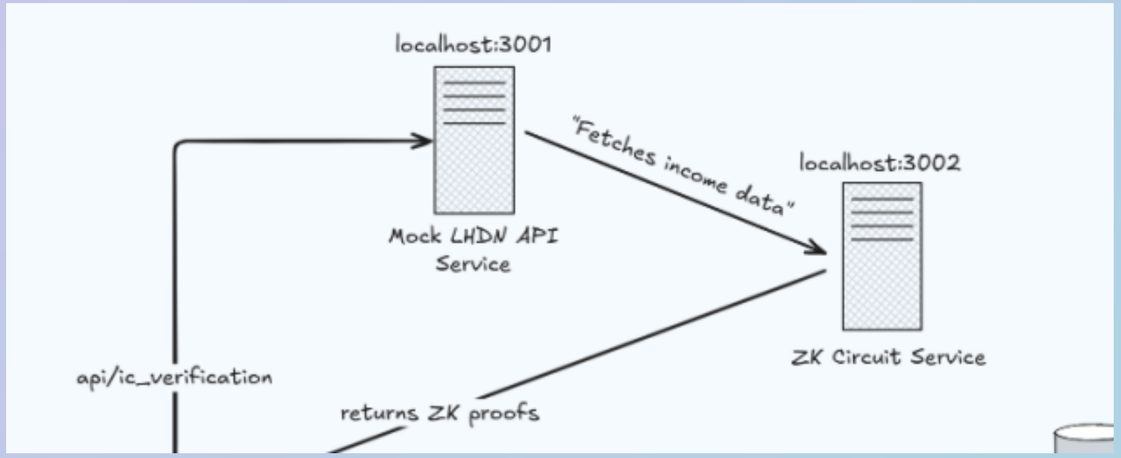
new improvement (smolagents + zk) (fyp2)



zero-knowledge proofs (zkps): Balancing Privacy and Verification

Objective

To allow citizens to prove their income bracket (e.g., B40-B1, M40, T20) for subsidy eligibility without revealing the exact income value



The Purpose of the Mock LHDN API

1. IC Validation
Performs rigorous format checking on the submitted Identity Card (IC) number.

2. Mock Income Lookup
Simulates government database functionality for development and testing environments, retrieving relevant financial information.

3. Data Security & Signing
Generates a digital signature (HMAC-SHA256) for the retrieved income data to ensure its integrity and authenticity.

ZKP Circuit Execution and Privacy Guarantee

ZKP Generation and Verification Workflow

Goal of ZK Circuit Service (Port 3002)

Generate cryptographic proof (using Groth16 zk-SNARKs) that demonstrates the citizen's income falls within a public bracket classification (B1-T2).

Input Preparation

Private Inputs (Hidden)

- Actual monthly_income (e.g., RM4,500)
- LHDN cryptographic signature

Public Inputs (Visible)

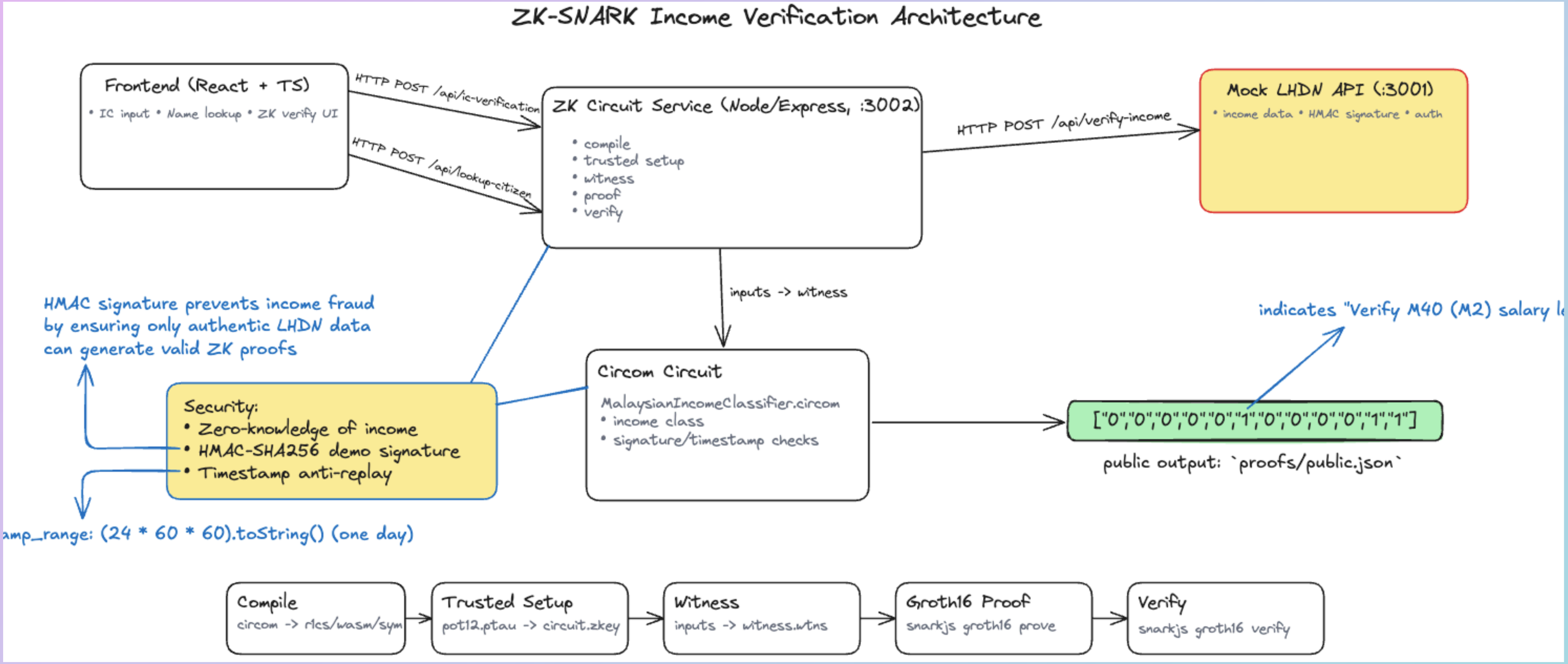
- IC hash identifier
- Malaysian income bracket thresholds (RM2,560, RM3,439, RM5,249...)

Core Circuit Logic (Witness Calculation)

1. **Signature Verification Module**
   Ensures the LHDN signature is mathematically valid and checks data freshness
2. **Income Threshold Comparison System**
   Uses LessThan components to evaluate private income vs public thresholds (e.g., 4500 < 5249 → B4 bracket)
3. **Security Gate**
   Conditional logic ensures classification output is valid only if LHDN signature verification succeeds

Proof Output and Verification

Groth16 Prover generates proof (Π) + public signal (income_bracket = B4). Verifier validates without accessing actual RM4,500.


ZK-SNARK Income Verification Architecture
Frontend (React + TS)
• IC input • Name lookup • ZK verify UI
HTTP POST /api/ic-verification
HTTP POST /api/lookup-citizen
ZK Circuit Service (Node/Express, :3002)
• compile
• trusted setup
• witness
• proof
• verify
HTTP POST /api/verify-income
Mock LHDN API (:3001)
• income data • HMAC signature • auth
inputs -> witness
HMAC signature prevents income fraud
by ensuring only authentic LHDN data
can generate valid ZK proofs
Security:
• Zero-knowledge of income
• HMAC-SHA256 demo signature
• Timestamp anti-replay
Circom Circuit
MalaysianIncomeClassifier.circom
• income class
• signature/timestamp checks
indicates "Verify M40 (M2) salary l
["0","0","0","0","0","1","0","0","0","0","1","1"]
public output: `proofs/public.json`
amp_range: (24 * 60 * 60).toString() (one day)
Compile
circom -> r1cs/wasm/sym
Trusted Setup
pot12.ptau -> circuit.zkey
Witness
inputs -> witness.wtns
Groth16 Proof
snarkjs groth16 prove
Verify
snarkjs groth16 verify

INPUTS: Data Required to Generate the Proof

The circuit requires a mix of private (hidden) and public (visible) inputs to perform its cryptographic calculations and ensure data authenticity [7] [8].

1. Private Inputs (Hidden from Verifiers)

These inputs constitute the sensitive information that remains concealed throughout the verification process (the Zero-Knowledge property) [2] [...].

Input Parameter	Description	Source
`monthly_income`	The citizen's actual monthly income in Ringgit Malaysia (RM) [8] [9]. [8]	
`signature`	An **HMAC signature** generated by the Mock LHDN API to cryptographically authenticate the income data [8]. [8]	
`verification_timestamp`	The timestamp indicating when the LHDN (Inland Revenue Board) verified and signed the income data [8]. [8]	

2. Public Inputs (Visible to Verifiers)

These inputs are necessary for the verification process but do not compromise the citizen's privacy [8] [10].

Input Parameter	Description	Source
`public_key`	The LHDN's cryptographic public key used by the Signature Verification Module to check the signature authenticity [8] [10]. [8] [10]	
`ic_hash`	A cryptographic **hash of the citizen's Identity Card (IC) number** used to bind the generated proof to a specific citizen identity [10] [11]. [10]	
`timestamp_range`	The maximum allowable age for the verification timestamp (e.g., 86400 seconds, representing one day) to prevent replay attacks and ensure data freshness [10]. [10]	

OUTPUTS: Verification Signals and Classification Result

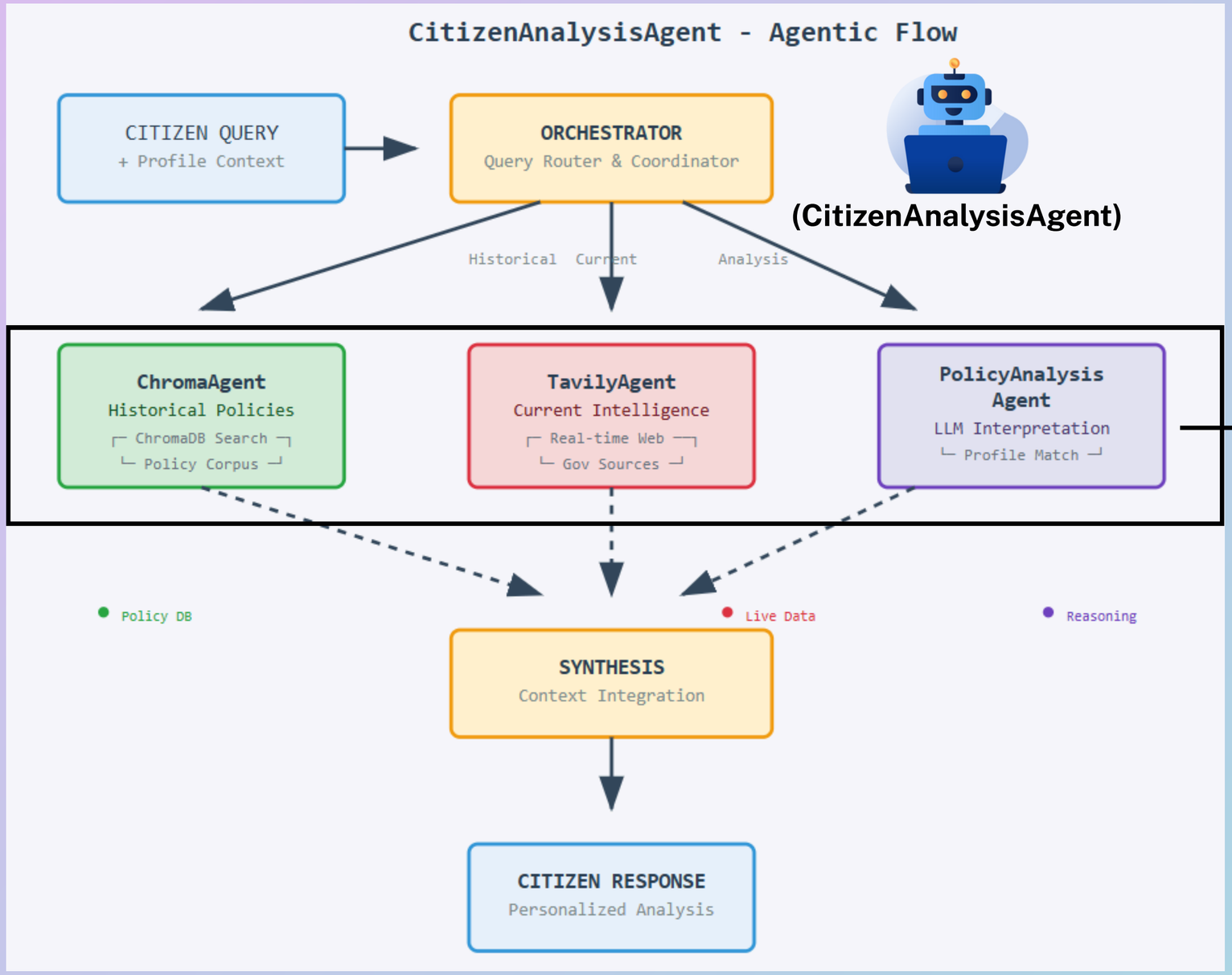
The circuit outputs signals that define the citizen's classification and confirm the integrity of the data used [12] [13].

Output Parameter	Description	Structure & Purpose	Source
`class_flags` [14]	The core classification result, a 10-element array representing the one-hot encoded income bracket [12] [13] .	Each position corresponds to brackets B1 through T2 (B1–B4, M1–M4, T1–T2). **Exactly one bit is set to '1'** to indicate the classification [12] [15] . [12] [13]	
`is_signature_valid`	Boolean flag indicating whether the LHDN cryptographic signature is mathematically valid [13] .	Confirms the origin of the income data [13] . [13]	
`is_data_authentic`	Boolean flag representing comprehensive data authenticity [13] .	Combines signature validity with timestamp and identity verification. If this flag is `0`, all classification flags are zeroed out, preventing valid proofs from unverified data [12] [16] . [13]	

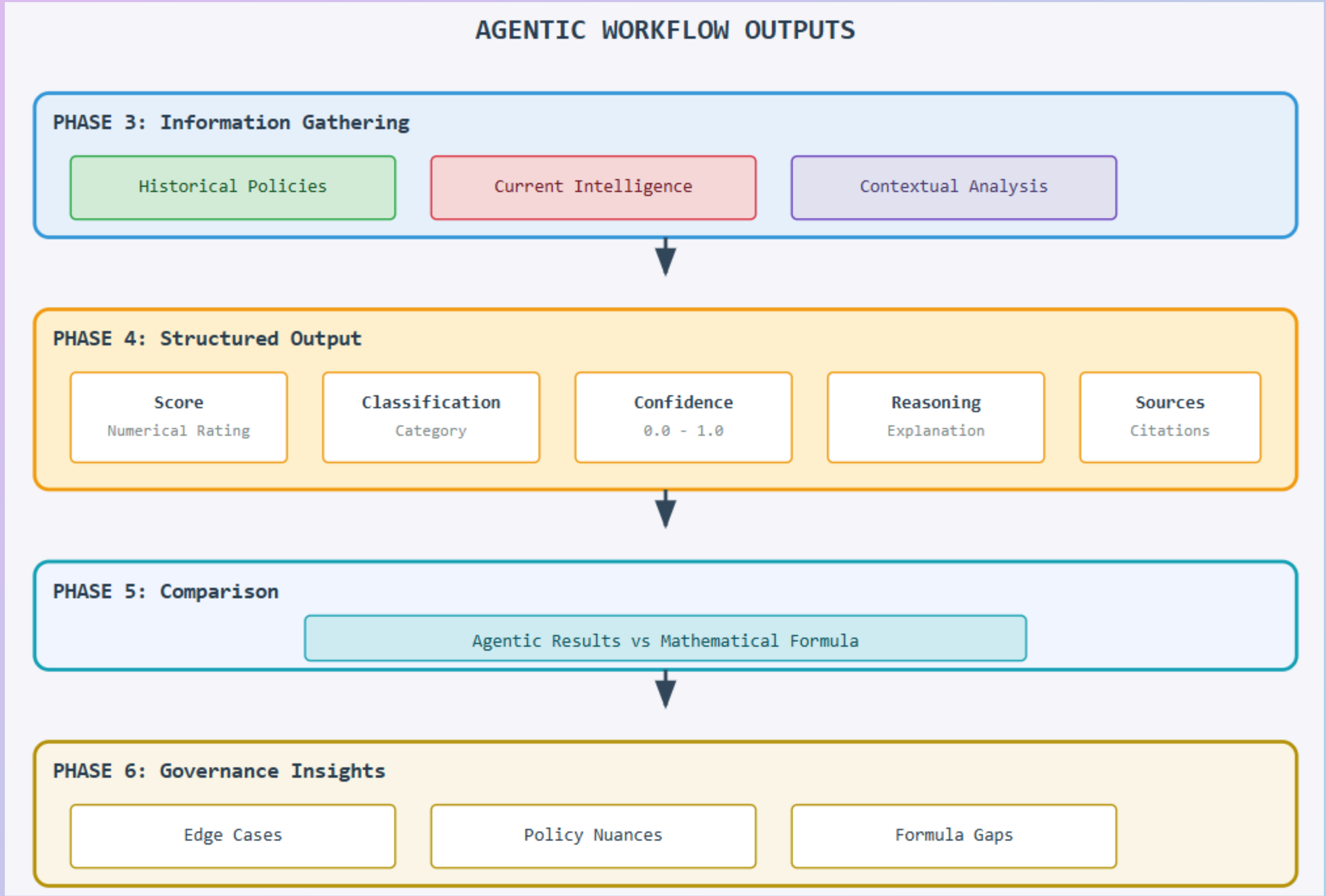
Example Output Interpretation [13] :

• If the output is `["0", "0", "0", "1", "0", "0", "0", "0", "0", "0", "1", "1"]`, it indicates:

agentic network for contextual policy reasoning (RAG Path)



in depth a bit~


AGENTIC WORKFLOW OUTPUTS
PHASE 3: Information Gathering
Historical Policies
Current Intelligence
Contextual Analysis
PHASE 4: Structured Output
Score
Numerical Rating
Classification
Category
Confidence
0.0 - 1.0
Reasoning
Explanation
Sources
Citations
PHASE 5: Comparison
Agentic Results vs Mathematical Formula
PHASE 6: Governance Insights
Edge Cases
Policy Nuances
Formula Gaps

implementation and testing results

Sepolia Testnet Explorer

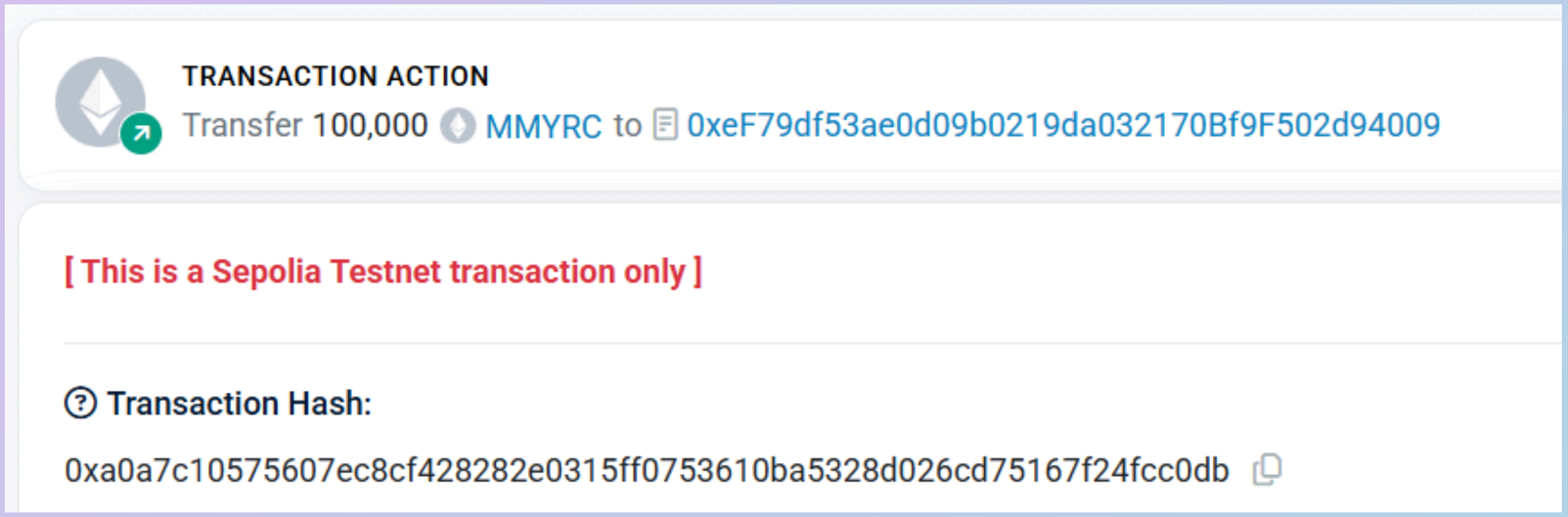
the project was developed as a working prototype, deployed on the Ethereum Sepolia Testnet using Visual Studio Code

successfully demonstrated the complete token distribution workflow. Total token supply capped at 1,000,000 MMYRC.

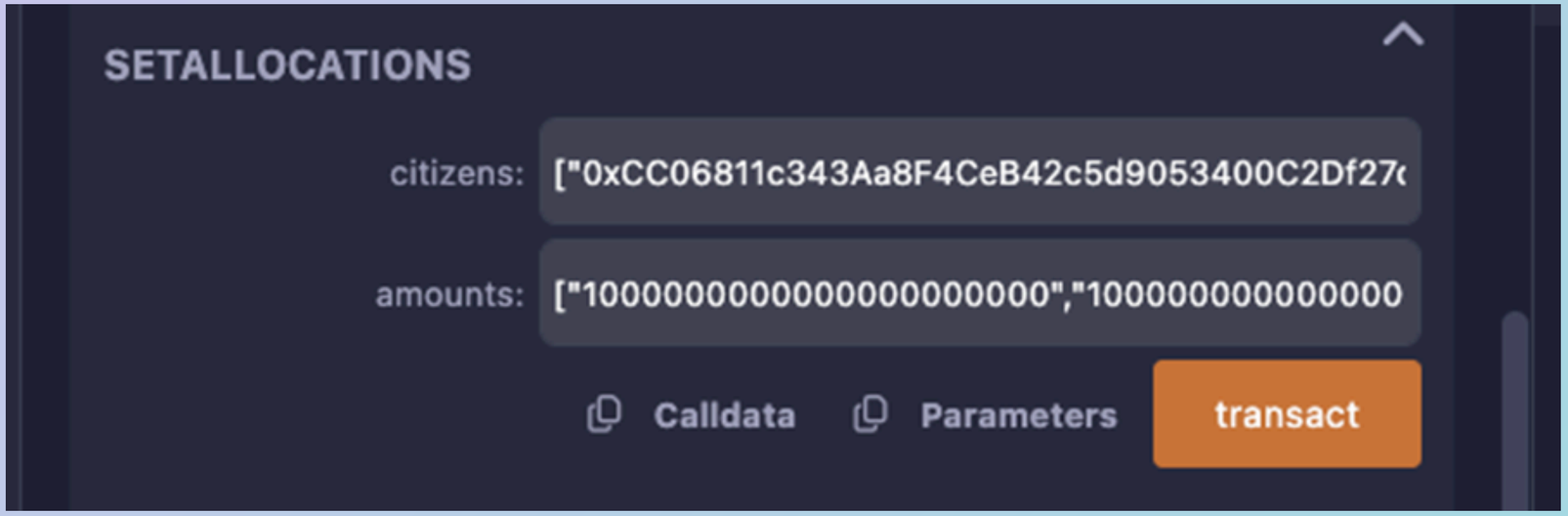
ZKP verification time is less than 1 second. The proof size is small (approx. 256 bytes) and remains constant regardless of input size, preserving computational efficiency

100,000 MMYRC were minted.

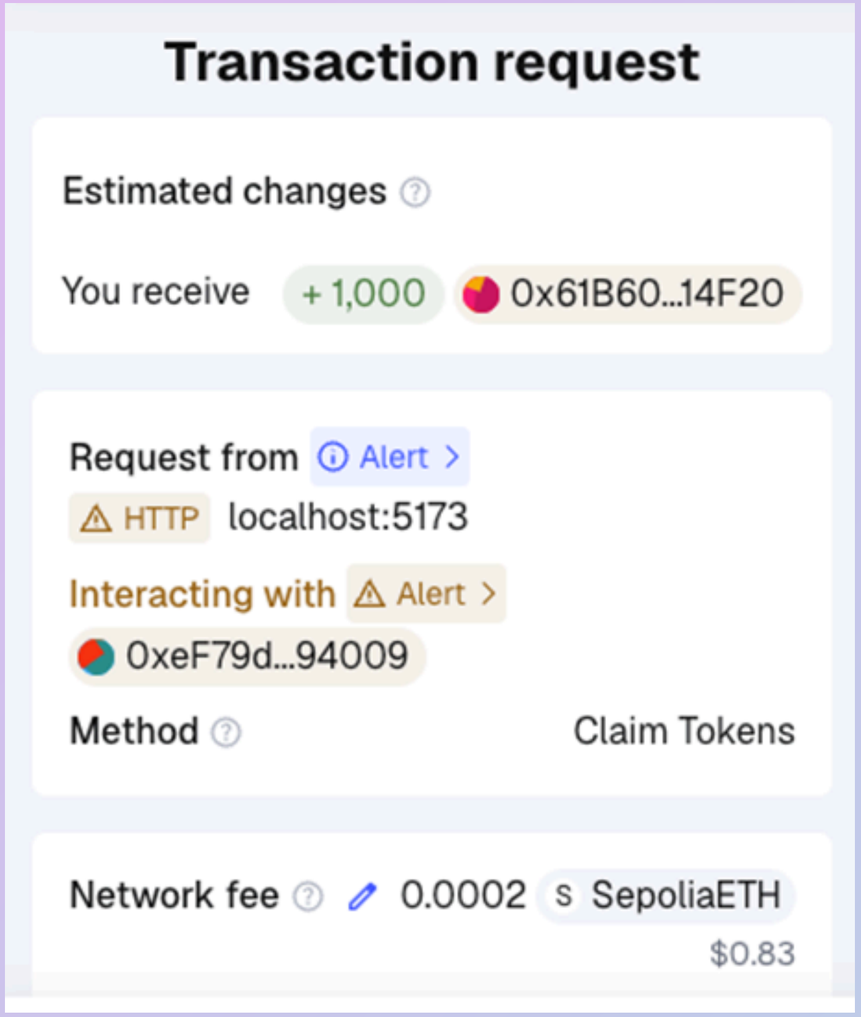
(10% of cap)

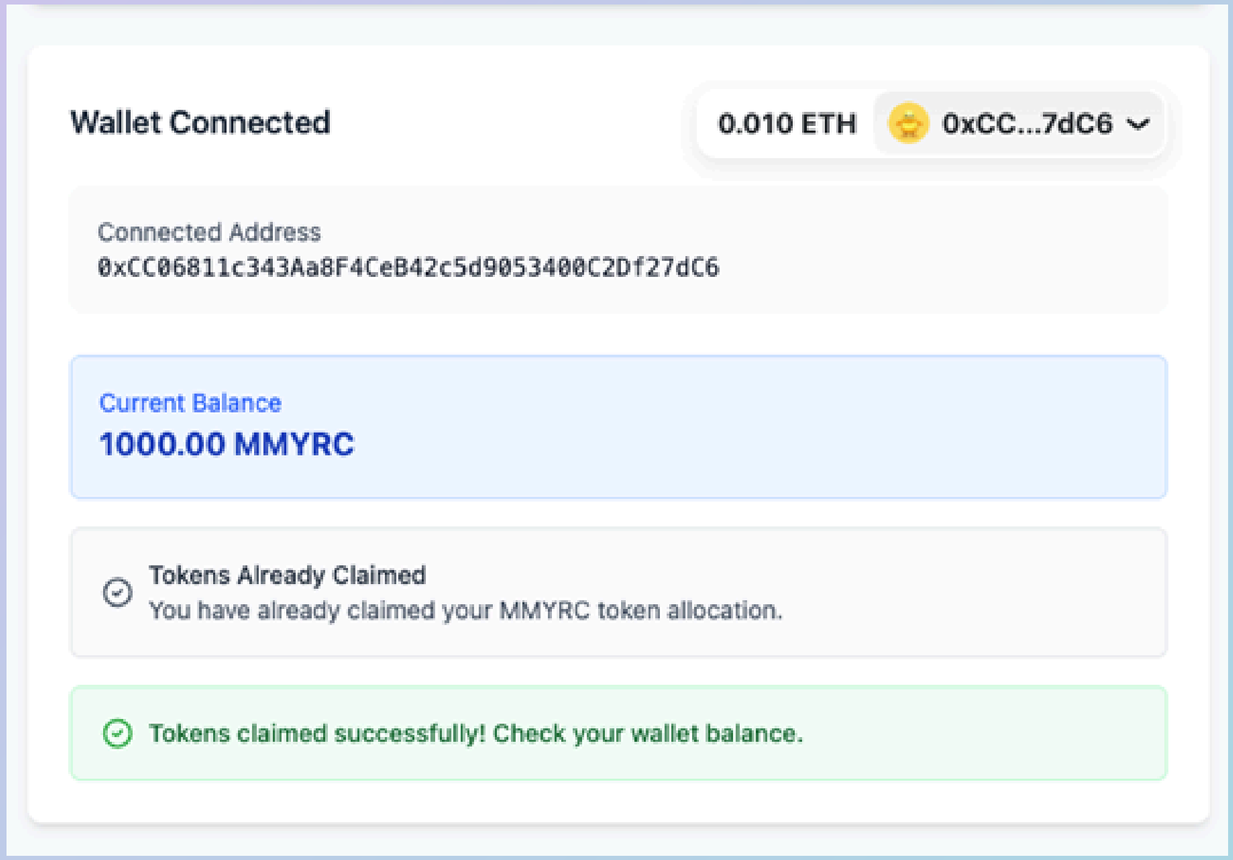


Eligible citizens were allocated 1,000 tokens each, demonstrating successful on-chain airdrops to verified eligible users



https://sepolia.etherscan.io/tx/0x85b28bae7f2450d03c5b262c0406d6d3bc804cdea3ea6bffbefeb3d739818623#eventlog


Transaction request
Estimated changes
You receive
+ 1,000
0x61B60...14F20
Request from
Alert
HTTP
localhost:5173
Interacting with
Alert
0xeF79d...94009
Method
Claim Tokens
Network fee
0.0002
SepoliaETH
$0.83


Wallet Connected
0.010 ETH
0xCC...7dC6
Connected Address
0xCC06811c343Aa8F4CeB42c5d9053400C2Df27dC6
Current Balance
1000.00 MMYRC
Tokens Already Claimed
You have already claimed your MMYRC token allocation.
Tokens claimed successfully! Check your wallet balance.

impact and significance

transparency & auditability

Every transaction is immutably recorded on a public ledger, minimizing fraud risk and strengthening public trust in government processes

citizen-centric design

The system balances transparency with individual privacy through the use of ZKPs, allowing citizens to verify eligibility without disclosing sensitive financial data

governance model

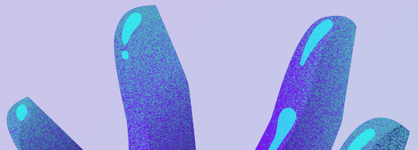
The project established a model for auditable, efficient, and citizen-centric public fund management by combining blockchain, AI-driven reasoning, and privacy-preserving technologies

contribution

- **Demonstrates feasibility of blockchain + AI subsidy distribution**
- **Lays foundation for future digital governance initiatives**

conclusion

The project successfully demonstrated the transformative potential of integrating blockchain and agentic frameworks into PADU to modernize subsidy distribution, achieving enhanced data integrity, transparency, and resilience.



recommendation

pilot deployment

Stage rollout across selected agencies

system integration

Link with national databases for seamless verification

security & trust

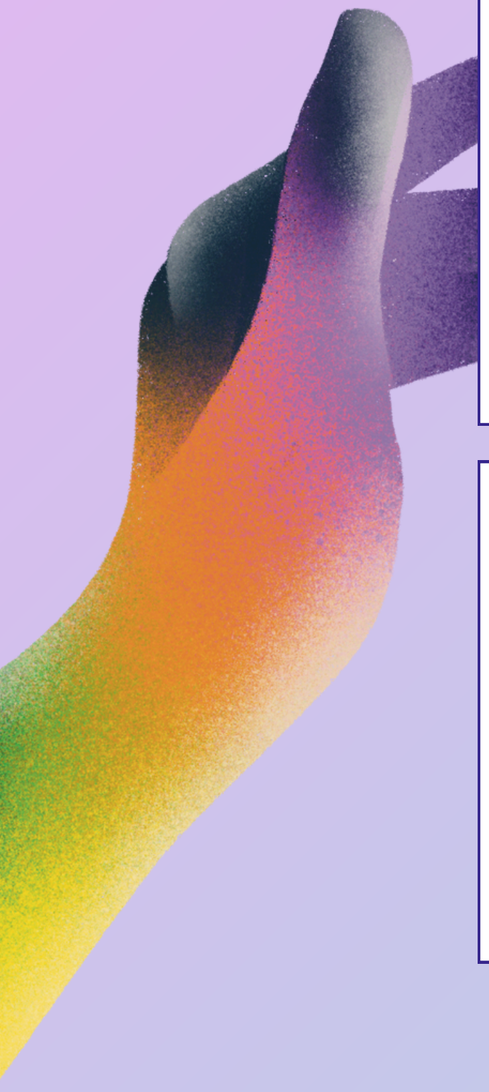
Perform smart contract audits

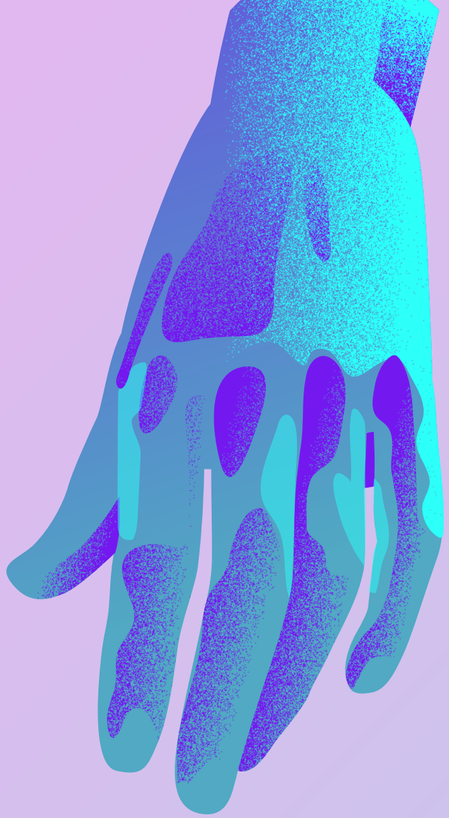
Explore broader use of ZKPs for citizen attributes

scalability & performance

Use Layer-2 solutions like zk-rollups

Optimize smart contracts & storage methods





thank you very much!