White Paper for Building a Local Currency System with Multi-Agent Technology

1. Executive Summary

This white paper proposes a next-generation local currency infrastructure based on Multi-Agent Systems (MAS). This system aims to evaluate and reward a wide range of resident activities, including actions, contributions, knowledge, enthusiasm, and trust networks, in real-time. It provides a decentralized economic platform that automatically designs and issues tokens and NFTs as rewards, enabling the transparent quantification of non-monetary contributions that were traditionally difficult to formalize.

Each agent in this system can monitor resident activities across both real-world environments and the metaverse, enabling the design of personalized incentives. This approach facilitates the creation of sustainable and autonomous regional economies, where even informal, intangible contributions are captured and rewarded. By leveraging the low transaction costs, high stability, and flexible smart contract capabilities of the Cardano blockchain, this system ensures a high level of reliability, tamper resistance, and transparency, making it particularly suitable for public-oriented local currencies.

Additionally, this system integrates advanced technologies such as Graph Neural Networks (GNNs) and Large Language Models (LLMs) to enable deep understanding of contribution behaviors and influence evaluation within social networks. This holistic approach allows for more accurate scoring of individual contributions and fairer reward distribution, thereby sustaining resident motivation and engagement over the long term.

A key feature of this approach is its integration with Domain-Specific Languages (DSLs) designed to work seamlessly with LLMs, enabling the no-code implementation of complex agent logic. This significantly reduces development costs and complexity, allowing non-engineers to easily design, update, and optimize agent behavior without sacrificing reliability.

This proposal goes beyond simple digitalization of local currencies, offering a comprehensive prototype for new regional social infrastructures that incorporate decentralized governance, cross-reality contribution evaluation, and trust score formation.

2. Background and Challenges

Local currencies have gained attention as a means to revitalize regional economies and strengthen community bonds. They have been introduced in various regions across the country as a tool for promoting local engagement. However, several structural challenges have become evident in their practical implementation:

- High Operational Costs and Administrative Burden: The processes of issuing, managing, and settling local currencies involve significant overhead, requiring substantial time and resources.

- Difficulty in Sustaining User Motivation and Engagement: Keeping residents continuously interested and motivated to use the currency over time remains a major hurdle.
- Lack of Quantifiable Economic and Social Impact: It is often challenging to accurately measure the economic and social outcomes of local currency initiatives.
- Limited Recognition of Non-Monetary Contributions: Conventional systems struggle to capture and appropriately reward non-financial contributions, such as volunteer work, knowledge sharing, and community support, leading to incomplete value assessments.
- Complexity in Logic Design and Maintenance: Traditional systems lack the flexibility to incorporate complex conditional logic and social factors, often relying heavily on specialized technical expertise for updates and maintenance.

Innovative Solutions through Web3 and Multi-Agent Systems (MAS)

To address these challenges, this proposal integrates Web3 technology with Multi-Agent Systems (MAS) to provide innovative solutions, including:

- Real-Time Scoring of Multi-Dimensional Contributions: Continuous assessment of residents' diverse activities, including actions, knowledge, and network contributions, allowing for more comprehensive and real-time value recognition.
- Automated and Optimized Token and NFT Distribution: Automatic design and distribution of rewards (tokens, NFTs) based on individual scores, reducing administrative overhead and increasing transparency.
- Infrastructure for Time-Series Analysis and Policy Impact Evaluation: Establishing a platform that enables real-time visualization and evaluation of the economic and social effects of regional policies and initiatives.

No-Code Implementation with DSL and LLM Integration

A standout feature of this approach is the use of Domain-Specific Languages (DSLs) integrated with Large Language Models (LLMs) to enable no-code implementation of complex agent logic. This innovation allows non-engineers to intuitively design, modify, and optimize agent behavior, significantly reducing development costs and improving system reliability. It also eliminates common maintenance bottlenecks, such as long lead times and high costs associated with expert-driven system updates, thereby dramatically increasing the flexibility and sustainability of local currency initiatives.

Cardano Blockchain as a Robust Foundation

The Cardano blockchain has been chosen as the foundation for this system due to its low transaction fees, high reliability, and flexible smart contract capabilities, making it particularly well-suited for public-oriented local currency systems. Its tamper-resistant, transparent ledger ensures high levels of trust and accountability, essential for the long-term success of regional economies.

Achieving Long-Term Sustainability and Quality Participation

Through this comprehensive approach to visualizing, validating, and rewarding contributions, this system aims to simultaneously improve the sustainability of local currency frameworks and enhance the quality of resident participation. By aligning economic incentives with community well-being, this platform seeks to redefine the role of local currencies in supporting vibrant, resilient local economies.

3. System Architecture

3.1 Core Structure

The proposed system is designed to autonomously execute a wide range of processes related to the issuance and management of local currencies through the integration of specialized agents, a time-series database, and the Cardano blockchain.

At the core of this architecture is the Orchestrator Agent, responsible for centrally managing the sequence of operations, data flows, trigger conditions, and error handling across multiple agents. This agent ensures the optimal operation of the entire system by coordinating complex workflows and handling exceptions effectively.

Furthermore, this proposal introduces a seamless integration between the Orchestrator Agent and LLM Agents, enabling the use of a Domain-Specific Language (DSL) to define complex processing flows. This approach allows for no-codeimplementation and updates, making it possible for non-engineers to flexibly design and adjust the control logic of the entire system, significantly reducing development time and improving maintainability.

The Metric Analyzer Agent is responsible for evaluating user behavior and incentive design. It quantitatively scores various forms of contribution, including local activities, event participation, knowledge sharing, and network building, storing these scores in XTDB. The accumulated scores are analyzed and referenced in real-time, facilitating the optimization of rewards such as currency, NFTs, and coupons, thus enabling personalized incentive designs for each user.

The Cardano Blockchain Integration Agent handles the issuance, distribution, and wallet integration of tokens and NFTs based on contribution scores, providing a transparent and tamper-resistant asset management infrastructure.

Additionally, the Cross-Reality Agent detects and evaluates user activities within virtual spaces like the Metaverse and Discord, integrating them with real-world initiatives. This ensures a consistent and visible contribution system across both virtual and physical spaces, allowing digital enthusiasm and influence to directly impact regional revitalization efforts.

Moreover, the View Agent provides a comprehensive dashboard for visualizing each user's contribution score trends, regional and temporal economic impacts, and behavioral distributions, supporting decision-making for evaluating and improving local policies.

As the foundational data platform for all these agents, XTDB manages input and output data, user contribution histories, and reward records in a time-series format. This centralized management approach enables state reconstruction and rollback analysis at any point in time, ensuring the reproducibility and reliability of the entire system.

3.2 Agent Details

3.2.1 Orchestrator Agent

The Orchestrator Agent serves as the central control unit for the entire system, coordinating dynamic interactions and state transitions between multiple agents based on processing flow definitions written in Domain-Specific Language (DSL).

In this proposal, the Orchestrator Agent is integrated with LLM Agents, enabling the generation and modification of DSL from natural language through a no-code approach. This innovation allows non-engineers, such as campaign managers and operations personnel, to intuitively describe and update processing sequences, execution conditions, priorities, and exception handling rules between agents without directly editing program code.

Additionally, the use of a DSL Checker enables automatic verification of both syntax and semantics in the written DSL, preventing design flaws and runtime errors before deployment. This approach significantly accelerates the development and operational cycles while greatly enhancing the reliability and maintainability of the overall system.

The Orchestrator Agent operates by interpreting and executing DSL scripts in real time, based on various events occurring within the system.

This DSL framework includes logic for trigger control, conditional branching, asynchronous processing, and state-dependent data integration, allowing the dynamic orchestration of collaborative workflows between multiple agents. This flexible control architecture enables the following key features:

- High Reactivity: Event-driven flow execution allows for immediate adaptation to changing conditions.
- Flexible Branching and Conditional Logic: Complex business rules and exception handling can be intuitively expressed through DSL.
- State-Based Coordination: The system can dynamically switch processes and integration targets based on the real-time state of agents or users.
- Scalability and Reusability: Each processing step is abstracted as a reusable DSL component, making it easy to customize and extend.

Furthermore, this DSL is integrated with LLM (Large Language Model) Agents, supporting automatic generation of formal processing flows from natural language. This means non-technical users can describe workflows like "Execute XXwhen YY happens," which are then instantly converted into structured DSL scripts.

In this way, the Orchestrator Agent serves as the central component of a DSL-driven multi-agent integration platform, balancing flexibility in configuration, transparency in operation, and scalability in control.

Below are examples of processing flows, demonstrating how DSL scripts can be generated and transformed by LLM agents.

(1) Basic Operations via DSL

DSL scripts are structured scripts that define the processing flows (processing graphs) between agents, written in formats like YAML, JSON, or custom syntaxes designed for specific use cases.

These scripts explicitly define various control structures, such as conditional branching (if), looping, and asynchronous calls (async call), based on trigger conditions like event-type or score-threshold. This approach allows for the flexible configuration of complex multi-agent collaboration logic while maintaining readability and reproducibility.

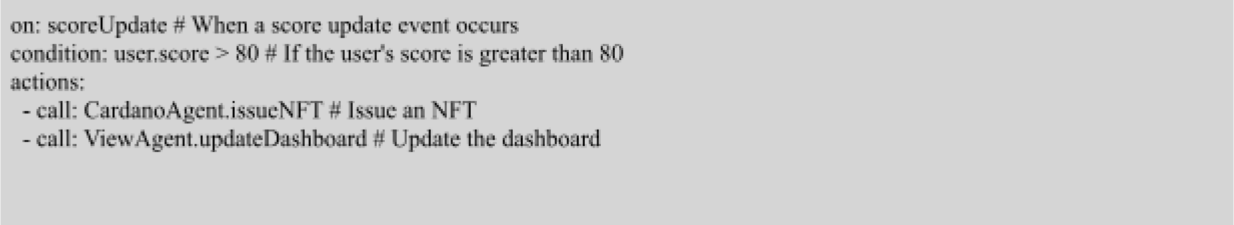
For example, the following DSL script could be generated via an LLM Agent for a process that triggers when a user's score is updated, and if the score exceeds 80, it issues an NFT and updates the dashboard accordingly:

<DSL Example>
The Orchestrator Agent then interprets and executes this DSL in real time.

(2) DSL-Based Execution of Agent Processing Flows

The Orchestrator Agent dynamically controls the sequence and execution conditions of processing flows, such as Metric Analyzer → Cardano Agent → View Agent, based on DSL scripts while communicating asynchronously with each agent through a message queue.

Within these DSL scripts, in addition to execution triggers and conditional branching, it is also possible to flexibly define control structures like parallel execution (parallel), wait conditions (waitUntil), rate limiting (debounce), and retry logic (retry). This allows for structured, asynchronous coordination of complex multi-agent interactions, enabling real-time process optimization and high operational flexibility.



```
on: scoreUpdate # When a score update event occurs
condition: user.score > 80 # If the user's score is greater than 80
actions:
  - call: CardanoAgent.issueNFT # Issue an NFT
  - call: ViewAgent.updateDashboard # Update the dashboard
```

For instance, consider the following multi-step agent interaction:

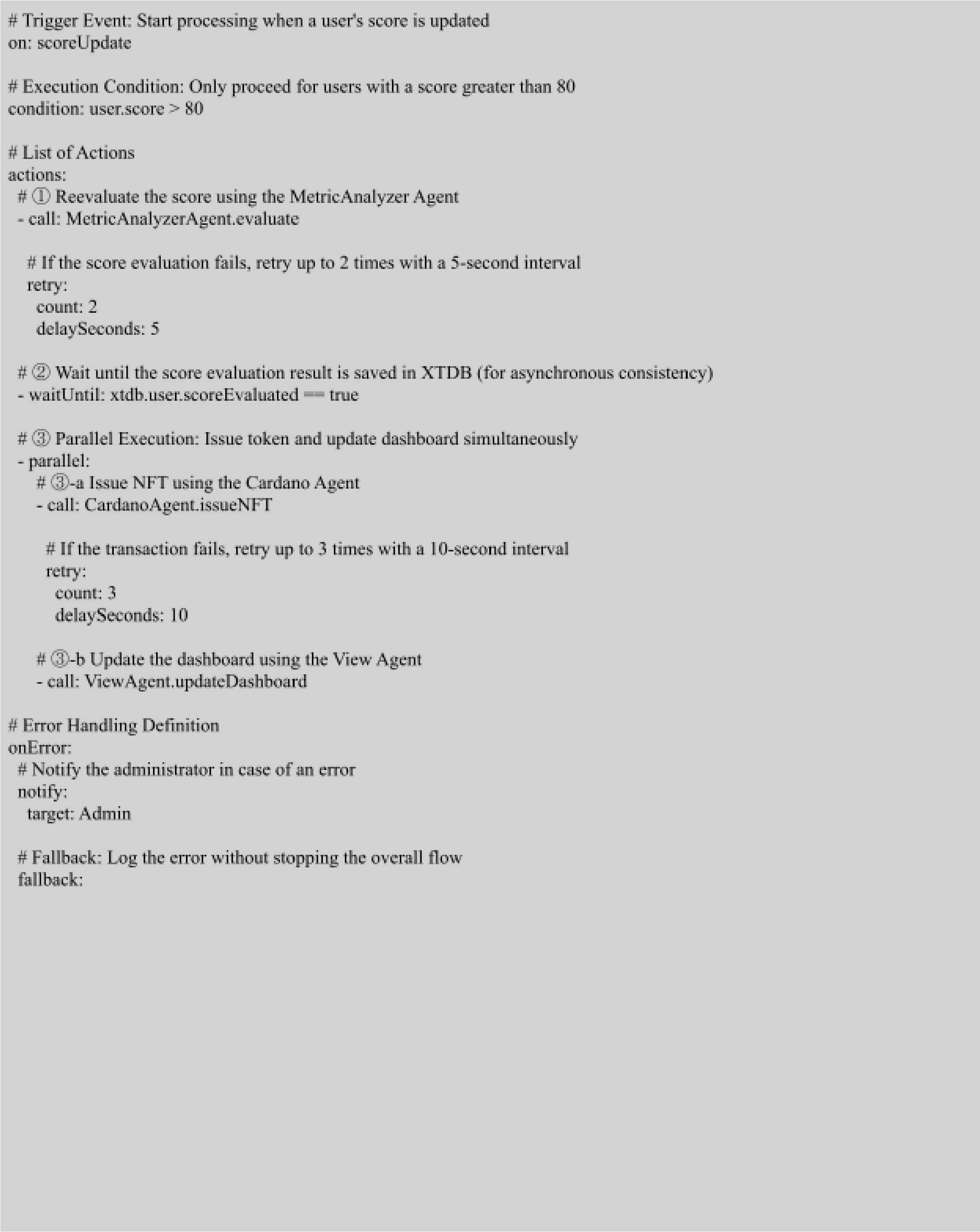
Scenario: When a user's score is updated, and the score exceeds 80, the following sequence of operations should be executed:

1. MetricAnalyzer Agent is called to reevaluate the score. If this evaluation fails, the system automatically retries up to 2 times with a 5-second interval between attempts.
2. Once the evaluation result has been successfully reflected in XTDB, the system waits for data synchronization to ensure consistency.
3. Two actions, NFT issuance and dashboard update, are then executed in parallel:
    - NFT Issuance: If a transaction error occurs, this action is configured to retry up to 3 times with a 10-second interval between attempts.
4. If any error occurs during the overall process, an alert is sent to the administrator, and the error details are logged. However, this fallback strategy is designed to log the error without halting the entire flow, ensuring continued system operation.

The following DSL script can be generated via the LLM Agent to define this workflow:

<DSL Example>

By defining processing flows as DSL (Domain-Specific Language) scripts, this approach enables a flexible and rapid response to specification changes. Additionally, the use of a



```
# Trigger Event: Start processing when a user's score is updated
on: scoreUpdate

# Execution Condition: Only proceed for users with a score greater than 80
condition: user.score > 80

# List of Actions
actions:
  # ① Reevaluate the score using the MetricAnalyzer Agent
  - call: MetricAnalyzerAgent.evaluate

    # If the score evaluation fails, retry up to 2 times with a 5-second interval
    retry:
      count: 2
      delaySeconds: 5

  # ② Wait until the score evaluation result is saved in XTDB (for asynchronous consistency)
  - waitUntil: xtdb.user.scoreEvaluated == true

  # ③ Parallel Execution: Issue token and update dashboard simultaneously
  - parallel:
      # ③-a Issue NFT using the Cardano Agent
      - call: CardanoAgent.issueNFT

        # If the transaction fails, retry up to 3 times with a 10-second interval
        retry:
          count: 3
          delaySeconds: 10

      # ③-b Update the dashboard using the View Agent
      - call: ViewAgent.updateDashboard

# Error Handling Definition
onError:
  # Notify the administrator in case of an error
  notify:
    target: Admin

  # Fallback: Log the error without stopping the overall flow
  fallback:
```

DSL Checker allows for automatic detection and validation of syntax errors and logical inconsistencies, ensuring high implementation quality and early error detection during the development phase.

Moreover, DSL definitions generated from natural language are easy to share with testers and stakeholders, facilitating alignment with specifications and supporting the creation of test plans. This makes it easier for non-engineers to understand, review, and suggest improvements to the processing logic, thereby promoting more efficient and transparent development and verification processes.

(3) Data Flow Control and XTDB Integration

It is possible to control data flows in conjunction with XTDB using a DSL (Domain-Specific Language). For example, consider the following data flow:

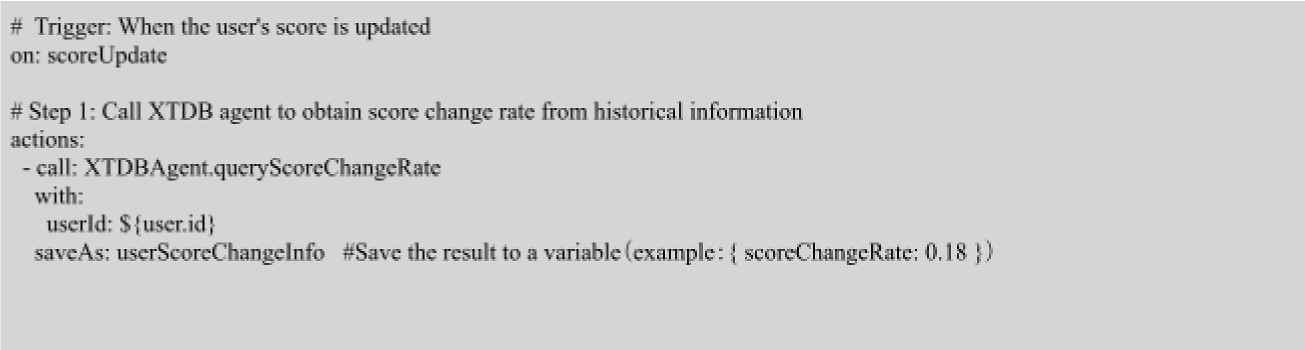
1. Trigger Event Detection:
   An XTDB agent detects a scoreUpdate event as a trigger, identifying that a user's score has been updated.

2. Historical Data Reference:
   Upon detecting this trigger, the XTDB agent references the historical data stored in XTDB. It checks if the score change represents a growth rate of 15% or more compared to the previous score. If this condition is met, the user is identified for further processing.

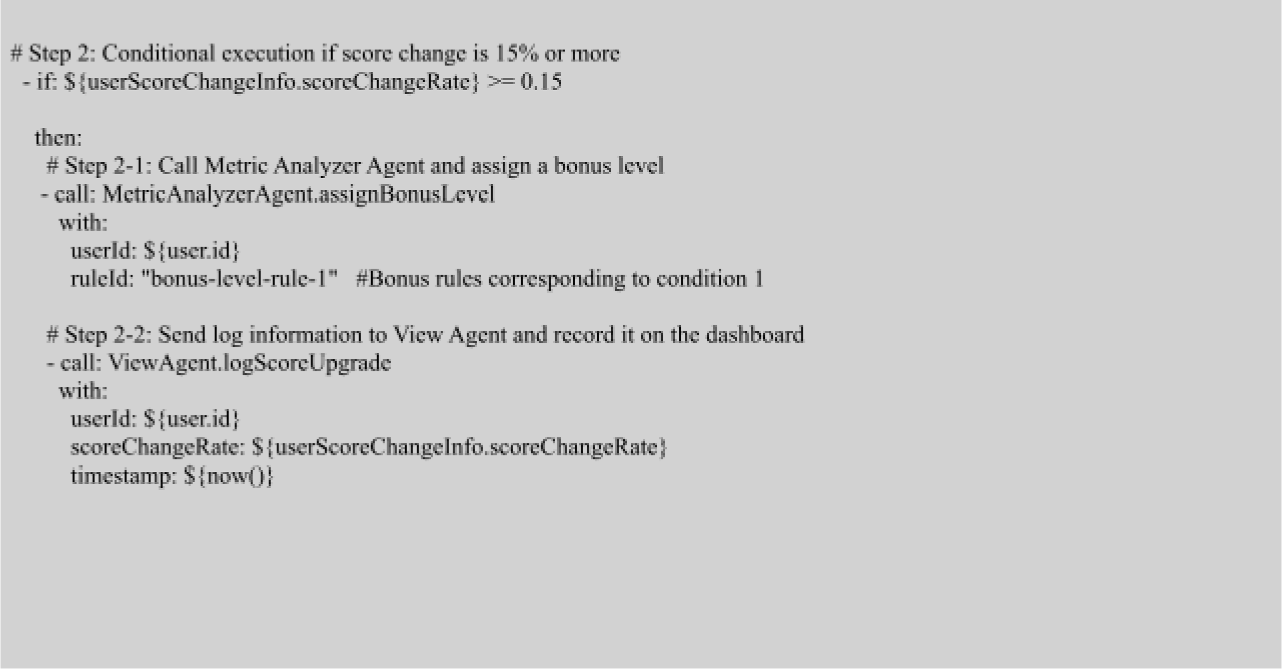
3. Metric Analysis:
   The identified user is then passed to the Metric Analyzer Agent, which automatically assigns a bonus level to the user based on a predefined Condition 1.

4. Dashboard Update:
   Finally, the ViewAgent is executed, updating the dashboard to reflect the score increase and the related processing details, ensuring that this information is logged and clearly presented for review.

Example DSL Generated by the LLM Agent:



```
#  Trigger: When the user's score is updated
on: scoreUpdate

# Step 1: Call XTDB agent to obtain score change rate from historical information
actions:
  - call: XTDBAgent.queryScoreChangeRate
    with:
      userId: ${user.id}
    saveAs: userScoreChangeInfo   #Save the result to a variable（example：{ scoreChangeRate: 0.18 }）
```



```
# Step 2: Conditional execution if score change is 15% or more
  - if: ${userScoreChangeInfo.scoreChangeRate} >= 0.15

    then:
      # Step 2-1: Call Metric Analyzer Agent and assign a bonus level
      - call: MetricAnalyzerAgent.assignBonusLevel
        with:
          userId: ${user.id}
          ruleId: "bonus-level-rule-1"   #Bonus rules corresponding to condition 1

      # Step 2-2: Send log information to View Agent and record it on the dashboard
      - call: ViewAgent.logScoreUpgrade
        with:
          userId: ${user.id}
          scoreChangeRate: ${userScoreChangeInfo.scoreChangeRate}
          timestamp: ${now()}
```

Data flows and database queries like these can also be consistently defined, generated, and utilized through a DSL (Domain-Specific Language). As described later, XTDB query definitions are also written in DSL, making the structure easy to understand even for non-engineers. This approach significantly enhances the overall transparency and maintainability of the system, allowing for straightforward modifications and reviews.

(4) Error and Exception Handling in DSL

Error and exception handling for each step can also be defined within the DSL, allowing for explicit fallback processes and alternative flows using the onError block. Additionally, by incorporating a DSL structure equivalent to try-catch, it is possible to integrate exception detection and handling directly into the processing flow. This design enables controlled responses to unexpected behaviors and partial failures, enhancing the overall robustness and reliability of the system.

system can be significantly improved.

For example, consider the following error handling process:
In the score retrieval step, the system attempts to call ScoreAgent.fetchLatestScore to obtain the latest score. However, if this step fails for any reason, the onError block automatically triggers the error handling flow.

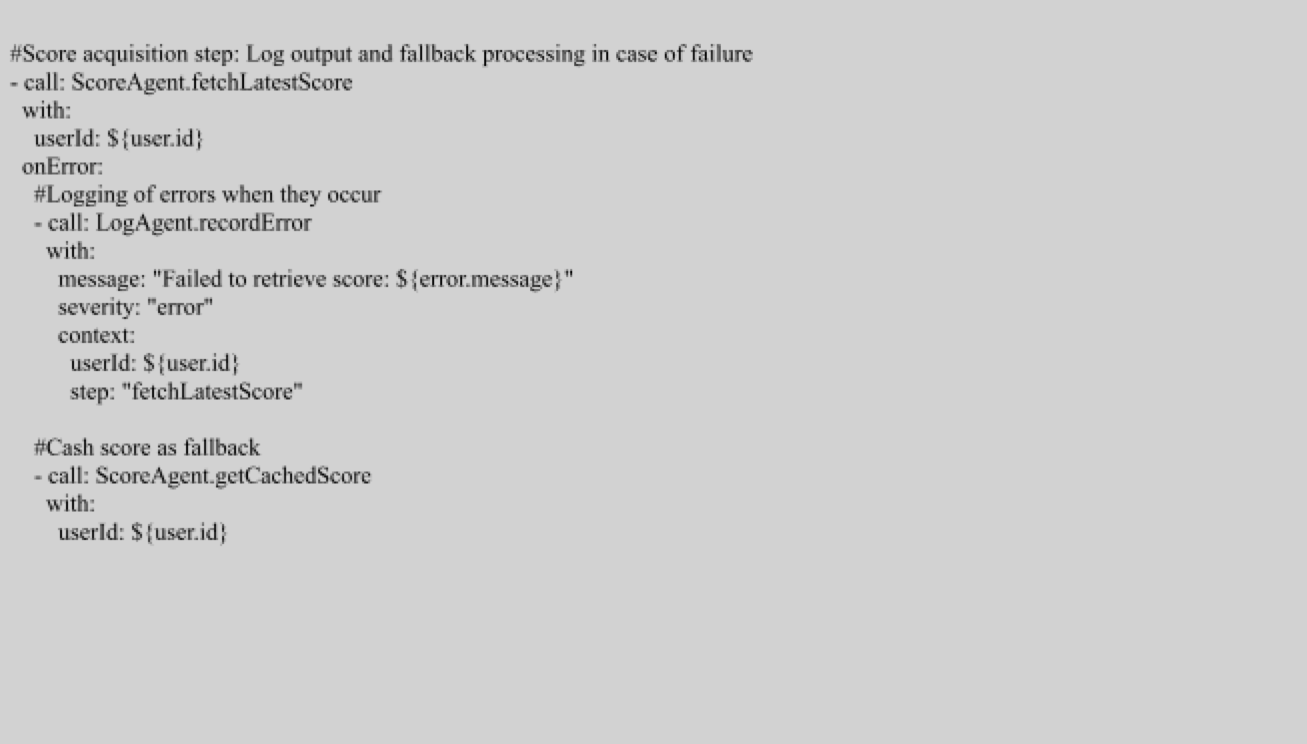
Specifically:

1. Error Logging:
   First, the error is recorded as an error log using LogAgent.recordError, capturing the error message and its context (including the user ID and step name).

2. Fallback Processing:
   Next, as an alternative measure, the system executes a fallback process by calling

ScoreAgent.getCachedScore to retrieve the cached score information, ensuring continuity even in the event of a primary failure.

The following DSL is generated using the LLM agent



```
#Score acquisition step: Log output and fallback processing in case of failure
- call: ScoreAgent.fetchLatestScore
  with:
    userId: ${user.id}
  onError:
    #Logging of errors when they occur
    - call: LogAgent.recordError
      with:
        message: "Failed to retrieve score: ${error.message}"
        severity: "error"
        context:
          userId: ${user.id}
          step: "fetchLatestScore"

    #Cash score as fallback
    - call: ScoreAgent.getCachedScore
      with:
        userId: ${user.id}
```

<Example DSL>
The generated DSL is interpreted and executed by the Orchestrator Agent. By defining error and exception handling directly within the DSL, fallback processes and error responses can be explicitly included as part of the overall business flow. This approach not only improves readability and reusability but also enables the system to handle unexpected errors through a structured, predefined error-handling framework. As a result, this robust design contributes to the stable operation of the service, reducing downtime and enhancing overall system reliability.

(5) Policy Switching and Mode Control

DSL scripts can be managed flexibly on a project or event basis, allowing for immediate policy changes. This structure enables the dynamic switching of processing logic based on different operational policies, such as "normal operation mode" and "campaign mode."

Settings can be controlled through variables and conditional blocks within the DSL, providing the ability to quickly respond to seasonal campaigns, emergency responses, or specific user segment targeting.

For example, consider the following flow:

First, the system checks the current operational mode (${config.mode}) and branches the processing accordingly:

1. Campaign Mode:

    - If the operational mode is set to "campaign", the system applies campaign-specific bonuses to the target users.
    - The CampaignAgent is called with CampaignAgent.applyBonusMultiplier, adding a 1.5x bonus to the user's score.
    - The details of this campaign processing, including the mode name and execution time, are then logged using ViewAgent.logCampaignAction, ensuring that the campaign actions are properly tracked and displayed on the dashboard.

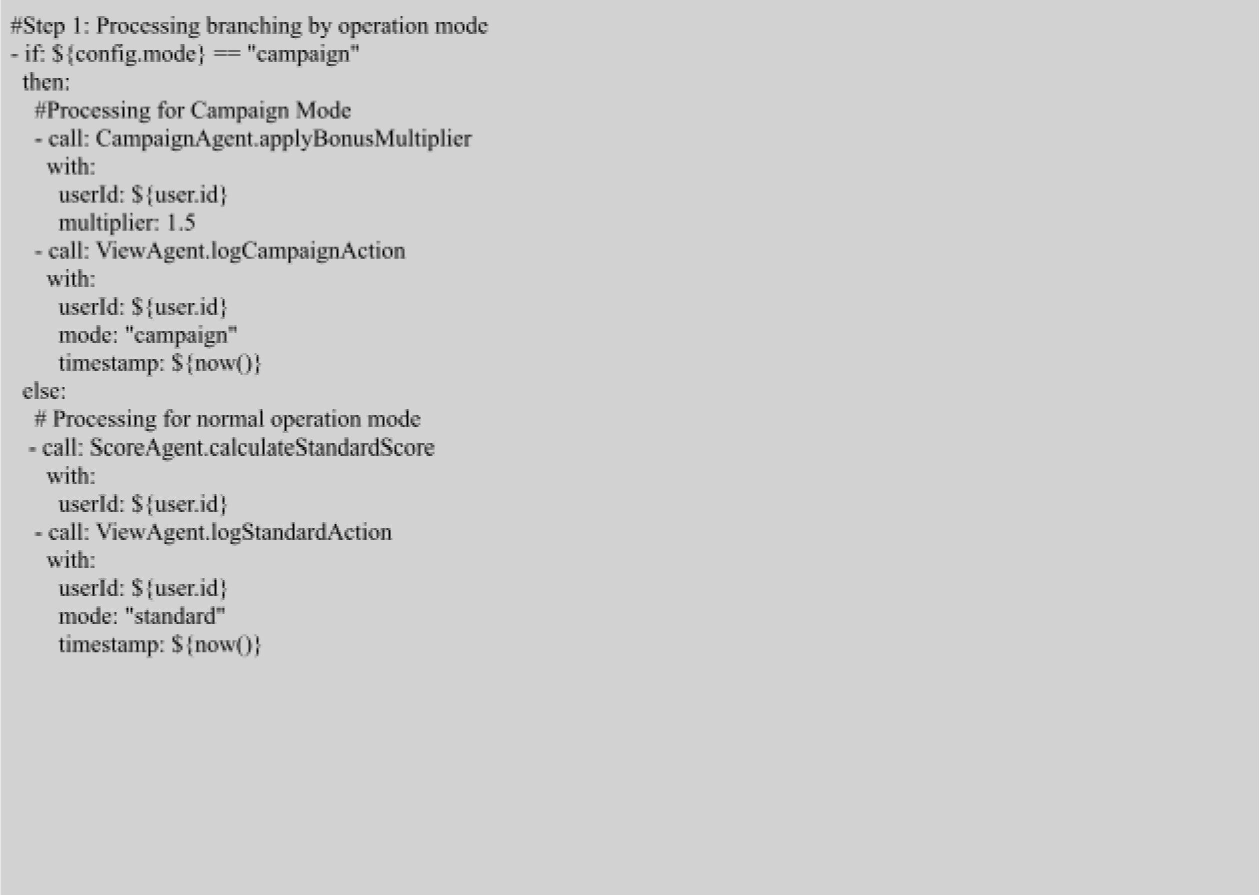
2. Normal Mode:

    - If the operational mode is anything other than "campaign" (i.e., normal operation mode), the system calculates the standard score using ScoreAgent.calculateStandardScore.
    - The result is then logged as a standard operation through ViewAgent.logStandardAction, maintaining accurate records of routine processing.

Example DSL Generated by the LLM Agent:

The generated DSL is interpreted and executed by the Orchestrator Agent. By utilizing DSL, policy changes can be handled at the DSL level, rather than requiring deep code modifications within the system. This approach minimizes operational overhead and risk, allowing for safe and rapid configuration updates. As a result, it significantly accelerates the overall strategy cycle, enhancing the responsiveness and agility of the system.

(6) Logging, Traceability, and Reproducibility



```
#Step 1: Processing branching by operation mode
- if: ${config.mode} == "campaign"
  then:
    #Processing for Campaign Mode
    - call: CampaignAgent.applyBonusMultiplier
      with:
        userId: ${user.id}
        multiplier: 1.5
    - call: ViewAgent.logCampaignAction
      with:
        userId: ${user.id}
        mode: "campaign"
        timestamp: ${now()}
  else:
    # Processing for normal operation mode
   - call: ScoreAgent.calculateStandardScore
      with:
        userId: ${user.id}
    - call: ViewAgent.logStandardAction
      with:
        userId: ${user.id}
        mode: "standard"
        timestamp: ${now()}
```

In this system, all DSL script versions and execution logs are designed to be recorded in XTDB, enabling precise post-hoc verification of which processing logic was executed and when.

Thanks to the bitemporal capabilities of XTDB, it is possible to distinguish between the execution time and the record update time, allowing the system to accurately reproduce past states. This makes it possible to trace and explain the causal relationships between specific strategies and their outcomes (e.g., score changes or user behavior) at the exact moment of execution.

This approach greatly enhances visibility into strategy effectiveness, ensures reproducibility, and strengthens accountability, making it possible to answer critical questions, such as:

- Which DSL version was used when a specific strategy was executed?
- What were the processing results for each affected user (e.g., whether the bonus was applied, the exact score change rate)?
- How did the targeted users' behavior and score trends evolve after the strategy was applied?

For example, if a campaign was executed during a specific period to "automatically assign bonus levels to users with a score change rate of 15% or more," this data can be used for:

- Verifying the exact DSL logic version in effect at the time

- Reviewing individual user processing results (e.g., bonus assignment status, score change rates)
- Analyzing the causal impact on user behavior and score trends post-campaign

These records serve as essential reference data for future strategy optimization and audit readiness.

All of this is made possible by XTDB's ability to distinguish between valid-time (the time when an event or fact was considered true) and transaction-time (the time when the record was physically stored in the database). This dual-timeline capability allows the system to accurately reproduce and explain past states, including the precise processing logic and execution context at a given point in time.

By versioning both processing logic and execution logs, and maintaining detailed records with time-axis context, the system can robustly support evidence-based strategy improvements, accountability, and reproducibility. This approach provides a powerful foundation for analyzing the causal relationships between specific strategies and their outcomes, making it possible to:

- Precisely track which logic was active at a given time
- Clearly explain the impact of past strategies, including user behavior and score changes
- Confidently reproduce historical system states for audit and optimization purposes

In this way, a DSL-integrated Orchestrator Agent becomes a critical component of the multi-agent system, enabling even non-engineers to edit control logic through structured rule descriptions. This significantly enhances the flexibility, readability, and reproducibility of the overall system.

3.2.2. XTDB

XTDB (eXtensible Time-based Database) serves as the core data infrastructure for this system, managing all agent input and output data, user contribution histories, reward logs, and transaction records in a time-series (bitemporal) format. It is not merely a storage medium but plays a critical role in ensuring reproducibility, auditability, and flexible query capabilities.

(1) Bitemporal Data Management (Valid Time / Transaction Time)

XTDB simultaneously manages Valid Time (the actual time an event occurred) and Transaction Time (the time the data was recorded in the database), capturing both "when the event happened" and "when it was known."

This dual-timeline approach allows the accurate reconstruction of the world state at any given historical point, making it possible to objectively verify whether decisions made in the past were reasonable based on the information available at that time.

In other words, it not only tracks the correctness of decisions but also provides a timeline-based rationale for why those decisions were correct, forming a critical foundation for reproducibility, accountability, and audit readiness.

An example of bitemporal data management might look like the following:

User ID	Score	Valid Time	Transaction Time
U123	60	2025-04-01T10:00:00	2025-04-01T10:05:00
U123	75	2025-04-01T10:00:00	2025-04-01T11:30:00 ←After correction
U123	90	2025-04-02T09:00:00	2025-04-02T09:01:00

This section provides an example of how a user's score history is recorded in XTDB.

- Valid Time represents the "actual time when the state was true" (e.g., a score of 75 that was valid as of April 1, 2025, 10:00).
- Transaction Time represents the "time when this information was registered or updated in the database"(e.g., the score 75 was registered later as a correction at 11:30).

Example 1: Reconstructing the "State of the World" at a Given Time

Suppose we want to reconstruct the state of the world at 10:10 on April 1, 2025.

- If the XTDB database is queried with Transaction Time = 2025-04-01T10:10, the recorded score for the user at that time was 60.
- This means that any decisions made at 10:10 (e.g., reward issuance, alert triggers) were based on the "world state"where the user's score was 60.

Even if the score was corrected later, XTDB can accurately reproduce the context in which the original decision was made, allowing for precise, time-based verification of past decisions.

Example 2: Analyzing the "Corrected Actual State" Including Retroactive Updates

Later, it is discovered that the user's actual score at 10:00 on April 1, 2025 was actually 75, not 60. This correction is registered in the database at 11:30 as a retroactive update:

- Valid Time: 2025-04-01T10:00
- Transaction Time: 2025-04-01T11:30

This means that the corrected record reflects the fact that the score 75 was actually valid at 10:00, even though this information was not registered until 11:30.

This bitemporal approach allows for reliable analysis of:

- Effectiveness of past strategies (e.g., bonus allocation based on score thresholds)
- Validation of decision accuracy (ensuring past decisions were reasonable given the available data at the time)
- Re-simulation of historical scenarios (e.g., analyzing how different score corrections might have impacted outcomes)

This capability significantly enhances the transparency, accountability, and reliability of the overall system.

(2) Contribution Score and Reward History Tracking

Data such as activity logs, score calculation results, and NFT issuance records from the Metric Analyzer Agent and Cardano Integration Agent are consistently and persistently stored in a time-series format. This approach enables precise tracking of when and how a user contributed, as well as the resulting score changes and reward issuances, on a per-user and per-period basis.

This mechanism provides a strong foundation for analyzing causal relationships, such as:

- Which contributions led to specific rewards?
- What behaviors were responsible for score increases over a given period?

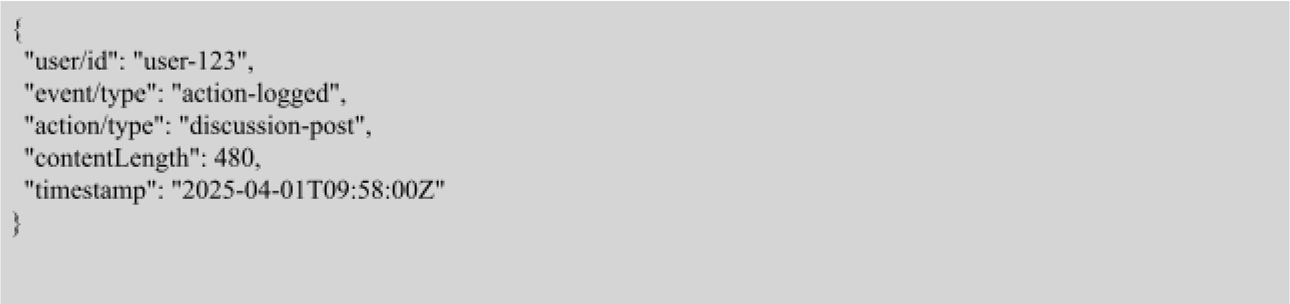
This enables robust evaluation of campaign effectiveness and clear visualization of user contributions.

Example:

① Tracking the Complete Flow of "Score Change → NFT Issuance" for a Single User

- At 9:58 AM on April 1, 2025, a user (ID: user-123) makes a "discussion post".
- This action is logged, capturing the key details, including the content length of

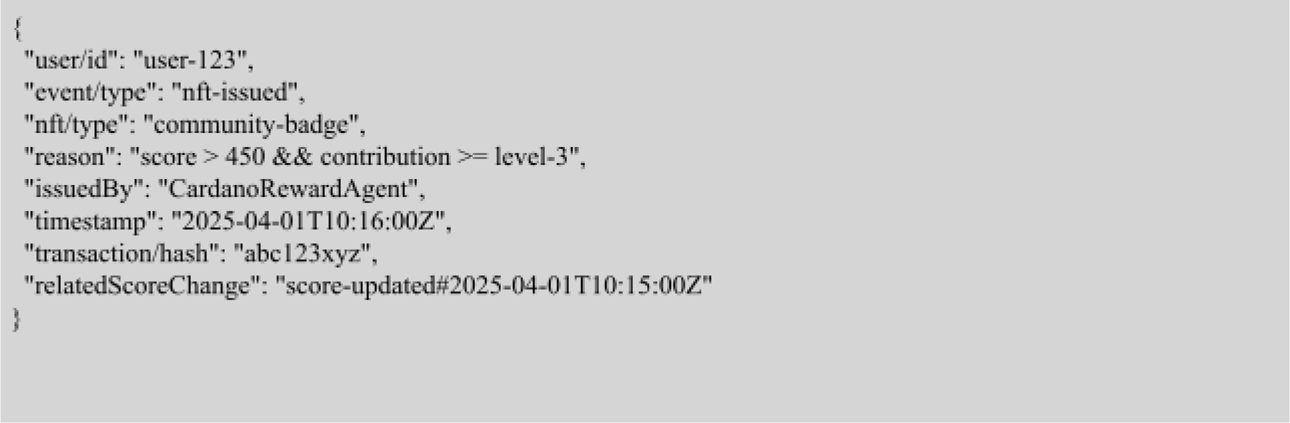
This posting activity is evaluated, and at 10:15 AM on April 1, 2025, the user's score increases from 420 to 480 points, reflecting a 14.2% increase.



```
{
  "user/id": "user-123",
  "event/type": "action-logged",
  "action/type": "discussion-post",
  "contentLength": 480,
  "timestamp": "2025-04-01T09:58:00Z"
}
```

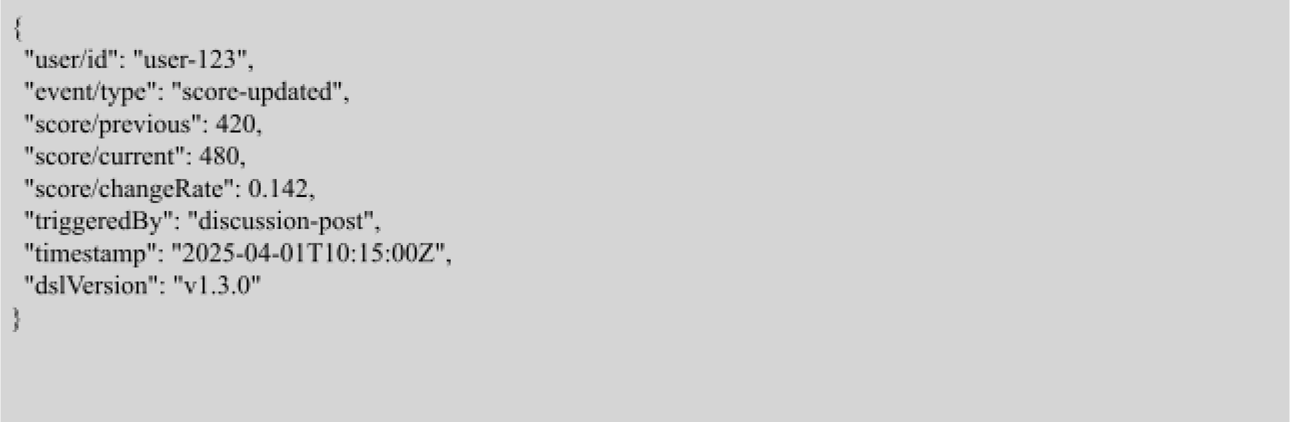
The trigger for this score update is explicitly linked to the previous "discussion-post" action. Additionally, this processing used DSL (Domain-Specific Language) version 1.3.0, clearly documenting the logic version applied in this context.

One minute later, the CardanoRewardAgent issues an NFT called "community-badge", based on the condition that the user's score has exceeded 450 and their contribution level has reached 3 or higher.



```
{
  "user/id": "user-123",
  "event/type": "nft-issued",
  "nft/type": "community-badge",
  "reason": "score > 450 && contribution >= level-3",
  "issuedBy": "CardanoRewardAgent",
  "timestamp": "2025-04-01T10:16:00Z",
  "transaction/hash": "abc123xyz",
  "relatedScoreChange": "score-updated#2025-04-01T10:15:00Z"
}
```

This NFT is directly linked to the previous score update, clearly establishing the relationship between the recent contribution and the reward issuance.

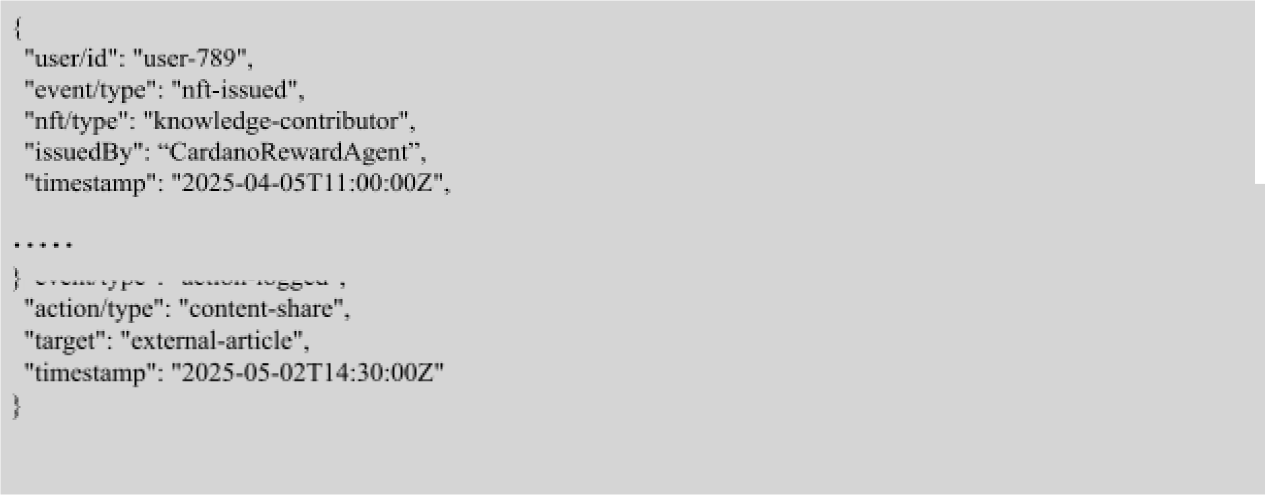


```
{
  "user/id": "user-123",
  "event/type": "score-updated",
  "score/previous": 420,
  "score/current": 480,
  "score/changeRate": 0.142,
  "triggeredBy": "discussion-post",
  "timestamp": "2025-04-01T10:15:00Z",
  "dslVersion": "v1.3.0"
}
```

② Verifying the Source of Score Increases Triggered by Specific Actions (with DSL Version Information)

It is recorded that a user (ID: user-456) performed a "content-share" action at 2:30 PM on May 2, 2025.

- The shared content type is "external-article", and this action has been logged in the system as "action-logged".
- The associated processing was handled using DSL version 1.3.0, clearly documenting the logic used for this contribution.



```
{
  "user/id": "user-789",
  "event/type": "nft-issued",
  "nft/type": "knowledge-contributor",
  "issuedBy": “CardanoRewardAgent”,
  "timestamp": "2025-04-05T11:00:00Z",

.....
} "event/type": "action-logged",
  "action/type": "content-share",
  "target": "external-article",
  "timestamp": "2025-05-02T14:30:00Z"
}
```

As a result of this "content-share" action, the user's score was evaluated and increased from 300 to 360 points, reflecting a 20% increase (+60 points).

The cause of this score change is explicitly identified as the "content-share" action, and the processing was executed using DSL version 1.3.0, clearly linking the contribution to the resulting score adjustment.

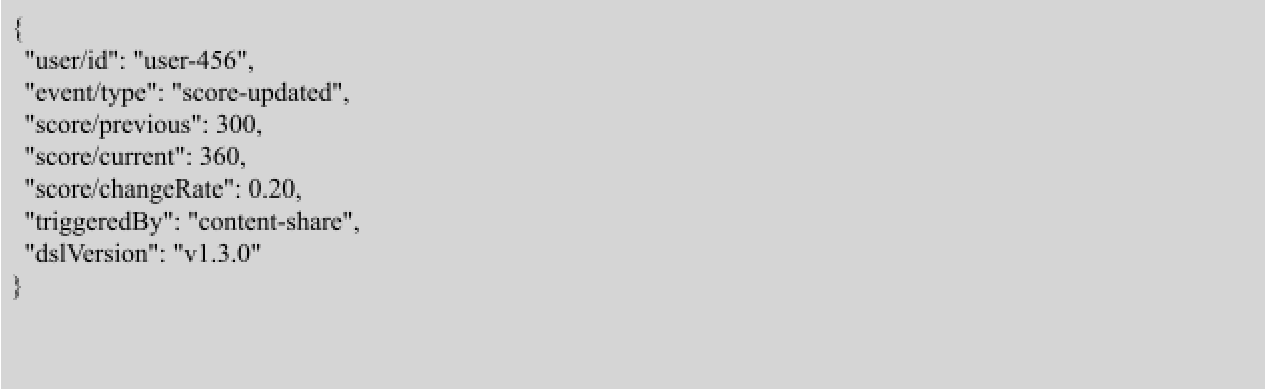
③ Clearly Explaining Which Score Changes, Actions, and Policy Applications Led to a Specific NFT Issuance

At 11:00 AM on April 5, 2025, an NFT called "knowledge-contributor" was issued to a user (ID: user-789) by the CardanoRewardAgent.

This issuance was triggered by the user making a meaningful knowledge contribution, which was quantitatively recognized as a score change, clearly linking the contribution to the resulting reward.

The reason for this NFT issuance is explicitly documented as "the score was updated and the processing was conducted using DSL version 1.3.0."

In other words, the score evaluation was performed in accordance with the official rule set (DSL), and the issuance was contingent on the user meeting specific contribution criteria.

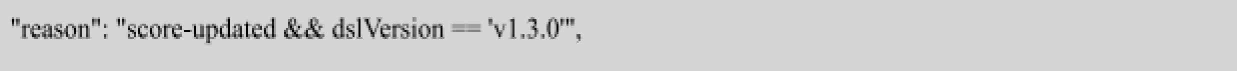


```
{
 "user/id": "user-456",
 "event/type": "score-updated",
 "score/previous": 300,
 "score/current": 360,
 "score/changeRate": 0.20,
 "triggeredBy": "content-share",
 "dslVersion": "v1.3.0"
}
```

The background for this NFT issuance is a high-contribution action in the form of a 950-word long-form post.

As a result of this post, the user's score increased from 700 to 790 points, representing an approximately 12.8% increase, aligning with the quantitative evaluation criteria.

④ Aggregating Score Changes for All Target Users Under Policy Version

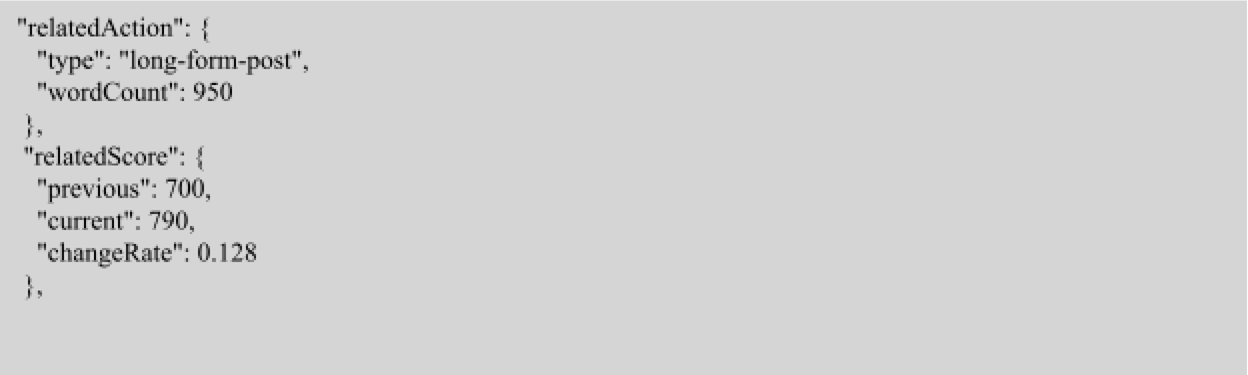


```
"reason": "score-updated && dslVersion == 'v1.3.0'",
```

v1.3.0 (Sample Aggregation)

The following dataset captures the score change logs for three users, each reflecting contributions evaluated under DSL version 1.3.0.

Specifically:



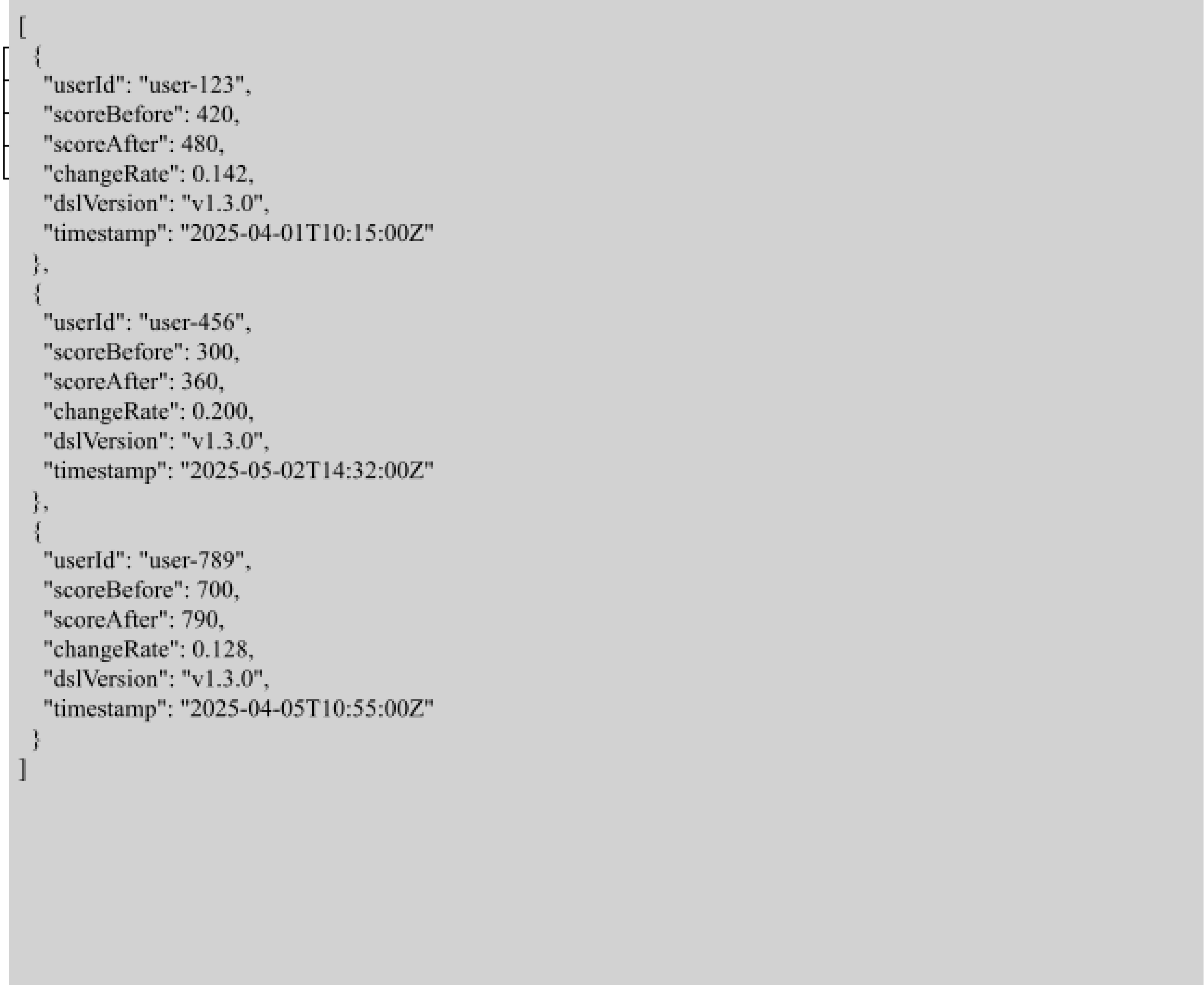
```
"relatedAction": {
   "type": "long-form-post",
   "wordCount": 950
 },
 "relatedScore": {
   "previous": 700,
   "current": 790,
   "changeRate": 0.128
 },
```

- user-123: On April 1, 2025, at 10:15, the score increased from 420 to 480, representing a 14.2% gain.
- user-456: On May 2, 2025, at 14:32, the score rose from 300 to 360, reflecting a 20.0% increase.

- user-789: On April 5, 2025, at 10:55, the score grew from 700 to 790, representing a 12.8% increase.

These records clearly indicate that each score update was processed in accordance with the v1.3.0 evaluation criteria.

Such score history aggregations can be retrieved using the following XTDB Datalog query.



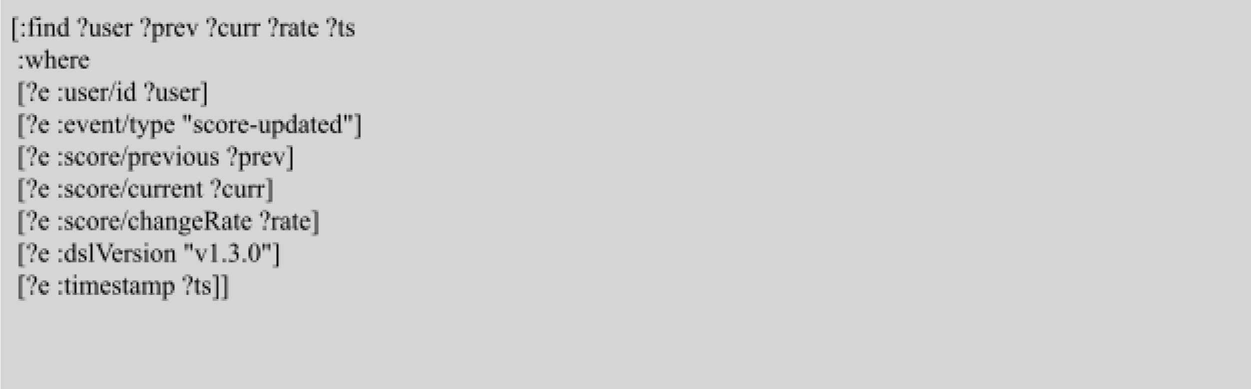
```
[
  {
   "userId": "user-123",
   "scoreBefore": 420,
   "scoreAfter": 480,
   "changeRate": 0.142,
   "dslVersion": "v1.3.0",
   "timestamp": "2025-04-01T10:15:00Z"
  },
  {
   "userId": "user-456",
   "scoreBefore": 300,
   "scoreAfter": 360,
   "changeRate": 0.200,
   "dslVersion": "v1.3.0",
   "timestamp": "2025-05-02T14:32:00Z"
  },
  {
   "userId": "user-789",
   "scoreBefore": 700,
   "scoreAfter": 790,
   "changeRate": 0.128,
   "dslVersion": "v1.3.0",
   "timestamp": "2025-04-05T10:55:00Z"
  }
]
```

The output results from this query are as follows.

(3) Flexible Information Extraction through Query-Based Processing

The XTDB Agent leverages the Datalog-based query mechanism provided by XTDB, enabling flexible information extraction based on complex conditions. For example, it can quickly execute aggregation and analysis tasks such as:

- Top 10 users with the highest score increases over the past 30 days



```
[:find ?user ?prev ?curr ?rate ?ts
 :where
 [?e :user/id ?user]
 [?e :event/type "score-updated"]
 [?e :score/previous ?prev]
 [?e :score/current ?curr]
 [?e :score/changeRate ?rate]
 [?e :dslVersion "v1.3.0"]
 [?e :timestamp ?ts]]
```

- Score history leading to specific NFT issuances
- Score change trends under a particular campaign version

This capability forms the backbone for integrating with Cross-Reality Agents and View Agents, providing real-time analytics to user interfaces and dashboards. This, in turn, supports data-driven decision-making and outcome visualization in operational environments.

Natural Language to Datalog Query Pipeline

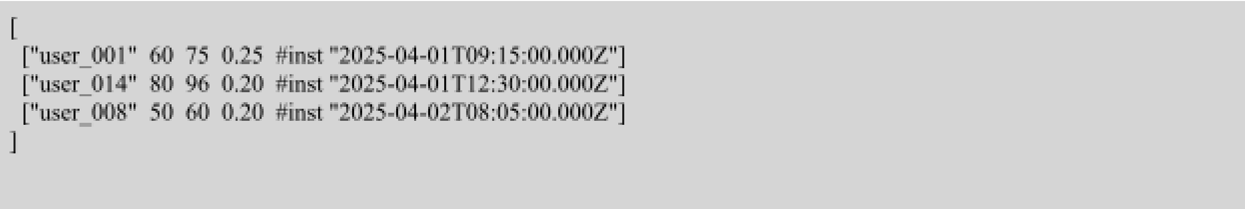
Datalog queries are automatically generated as an intermediate format from the DSL created by the LLM Agent based on natural language input. By incorporating this DSL layer, users can construct complex, code-free queries, while the DSL checker ensures the syntactic and semantic validity of the generated queries, providing a reliable processing framework.

This end-to-end pipeline consists of:

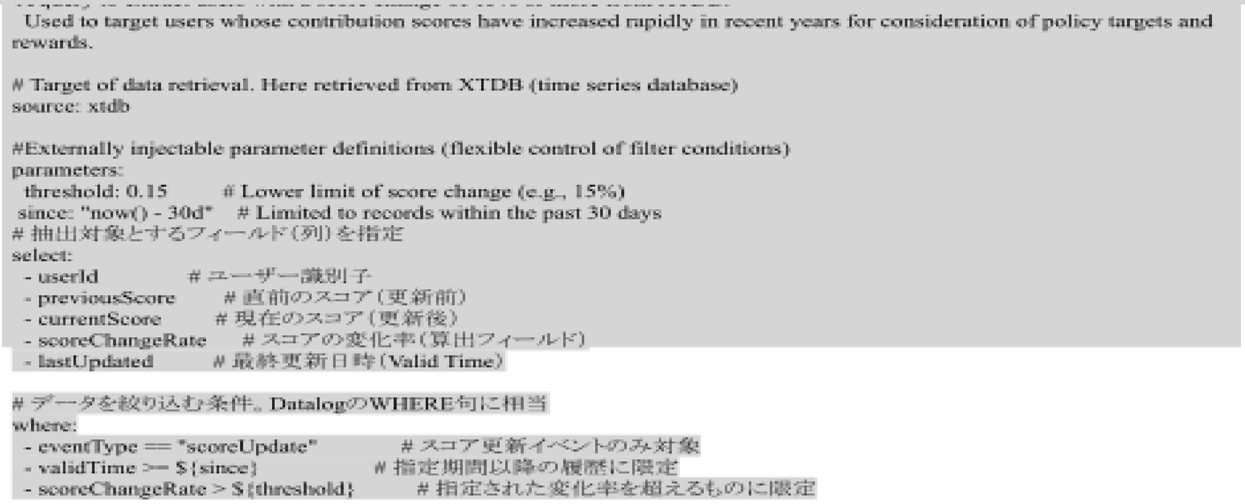
- Natural Language Input → DSL Generation → Datalog Conversion → Validation → Execution

This approach creates a seamless workflow that balances operational productivity with overall system reliability.

Example Use Case:

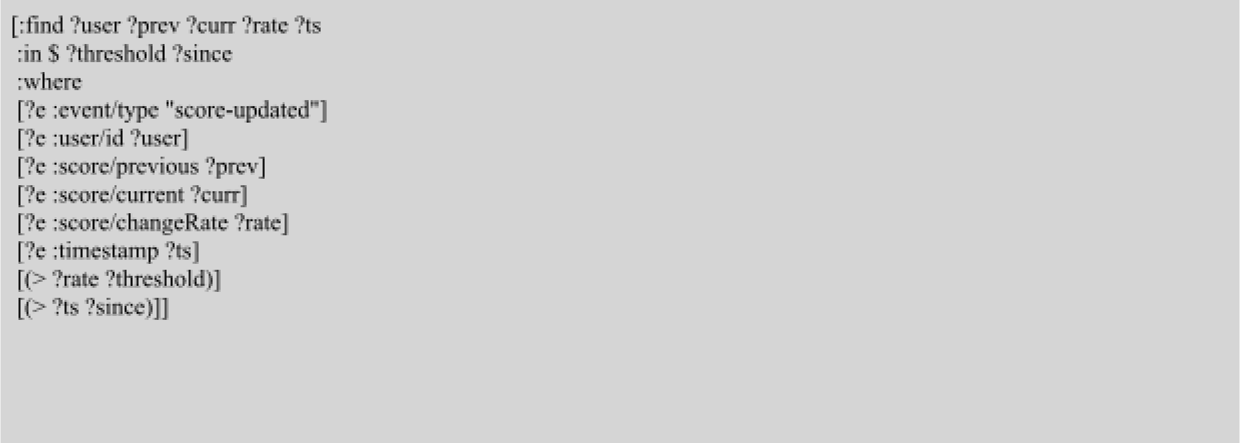


```
[
 ["user_001"  60  75  0.25  #inst "2025-04-01T09:15:00.000Z"]
 ["user_014"  80  96  0.20  #inst "2025-04-01T12:30:00.000Z"]
 ["user_008"  50  60  0.20  #inst "2025-04-02T08:05:00.000Z"]
]
```



```
  Used to target users whose contribution scores have increased rapidly in recent years for consideration of policy targets and
rewards.

# Target of data retrieval. Here retrieved from XTDB (time series database)
source: xtdb

#Externally injectable parameter definitions (flexible control of filter conditions)
parameters:
  threshold: 0.15        # Lower limit of score change (e.g., 15%)
 since: "now() - 30d"    # Limited to records within the past 30 days
# 抽出対象とするフィールド（列）を指定
select:
  - userId            # ユーザー識別子
  - previousScore       # 直前のスコア（更新前）
  - currentScore        # 現在のスコア（更新後）
  - scoreChangeRate      # スコアの変化率（算出フィールド）
  - lastUpdated         # 最終更新日時（Valid Time）

# データを絞り込む条件。DatalogのWHERE句に相当
where:
  - eventType == "scoreUpdate"           # スコア更新イベントのみ対象
  - validTime >= ${since}              # 指定期間以降の履歴に限定
  - scoreChangeRate > ${threshold}         # 指定された変化率を超えるものに限定
```

For instance, if you want to generate a query to "extract users with a score increase of 15% or more from XTDB for potential inclusion in upcoming campaigns or reward evaluations", the DSL generated through the LLM Agent would look like the following:



```
[:find ?user ?prev ?curr ?rate ?ts
 :in $ ?threshold ?since
 :where
 [?e :event/type "score-updated"]
 [?e :user/id ?user]
 [?e :score/previous ?prev]
 [?e :score/current ?curr]
 [?e :score/changeRate ?rate]
 [?e :timestamp ?ts]
 [(> ?rate ?threshold)]
 [(> ?ts ?since)]]
```

The Datalog query for XTDB generated by this DSL is as follows

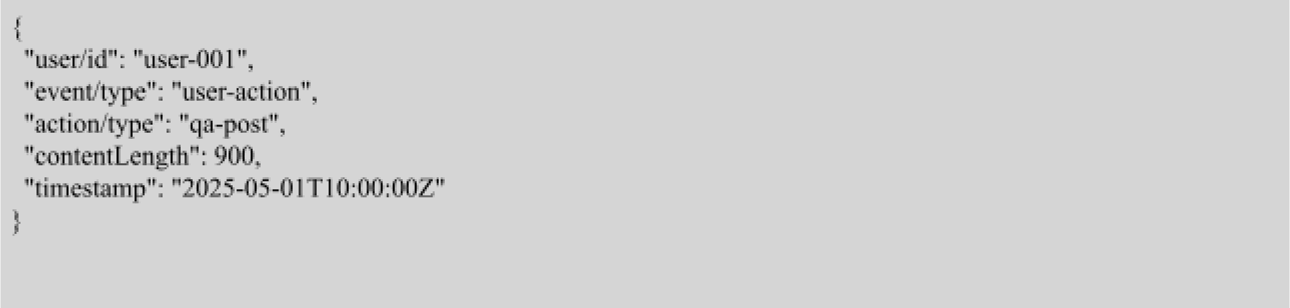
The query results for this query are as follows

(4) Asynchronous Agent Coordination and Data Sharing

In this system, each agent interacts asynchronously through indirect data sharing via XTDB, resulting in a loosely coupled architecture with minimal inter-agent dependencies. This design allows agents to read and write data according to their specific roles without the need for direct communication, promoting independent processing.

Additionally, the Orchestrator Agent acts as a central control mechanism, using XTDB updates (writes) and state reads as triggers for branching and executing subsequent processing. This significantly enhances the system's overall flexibility and scalability, making it possible to add new agents or processing logic without disrupting existing components.

Example: Reward Processing Flow Based on User Actions with XTDB

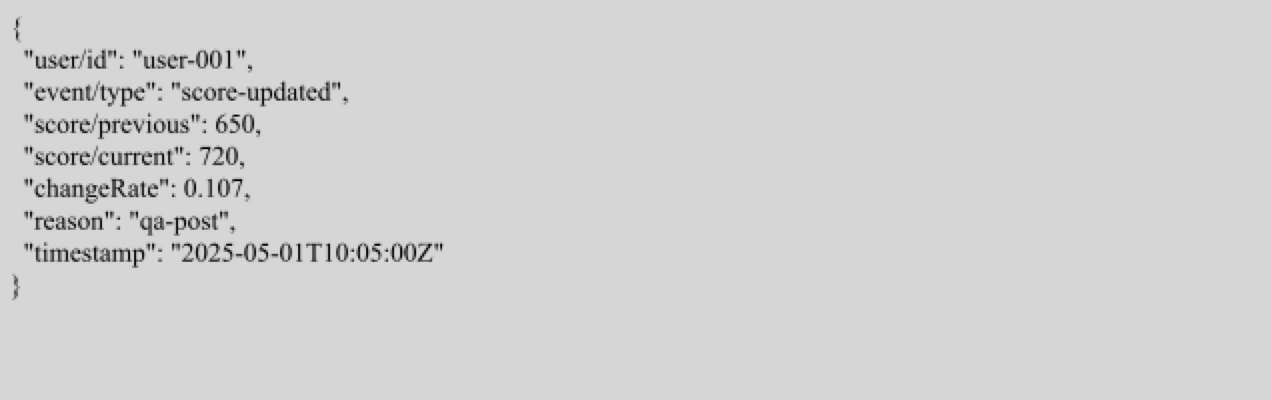


```
{
 "user/id": "user-001",
 "event/type": "user-action",
 "action/type": "qa-post",
 "contentLength": 900,
 "timestamp": "2025-05-01T10:00:00Z"
}
```

1 User Performs an Action
The UserActionAgent writes user action logs, such as long-form posts or Q&A participation, to XTDB. For example:

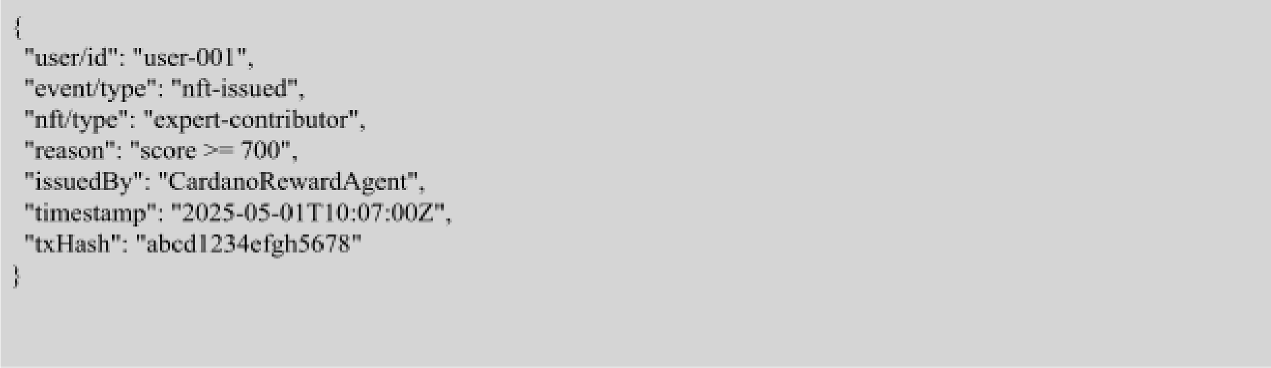
② Orchestrator Agent Triggered by XTDB Write
The OrchestratorAgent monitors the appearance of the above event in XTDB. Upon detecting that the specified conditions (e.g., the post is 800 characters or more) are met, it triggers the invocation of the MetricAnalyzerAgent.

③ Score Calculation by MetricAnalyzerAgent
The MetricAnalyzerAgent calculates the user's score based on the provided action details and records the result back into XTDB.



```
{
 "user/id": "user-001",
 "event/type": "score-updated",
 "score/previous": 650,
 "score/current": 720,
 "changeRate": 0.107,
 "reason": "qa-post",
 "timestamp": "2025-05-01T10:05:00Z"
}
```

④ Cardano Integration Agent Issues NFT Based on Score Change



```
{
  "user/id": "user-001",
  "event/type": "nft-issued",
  "nft/type": "expert-contributor",
  "reason": "score >= 700",
  "issuedBy": "CardanoRewardAgent",
  "timestamp": "2025-05-01T10:07:00Z",
  "txHash": "abcd1234efgh5678"
}
```

The CardanoRewardAgent monitors score-updated events in XTDB and, if the score exceeds the predefined threshold, triggers the issuance of an NFT.
⑤ ViewAgent Updates the Dashboard
The ViewAgent aggregates the relevant user actions, score changes, and NFT issuance history from XTDB, displaying this information on the UI.

In this way, each agent detects, writes, and monitors state changes through XTDB without direct communication, enabling asynchronous coordination. This approach results in a loosely coupled and highly extensible architecture.

(5) Data Validation and Audit Compliance

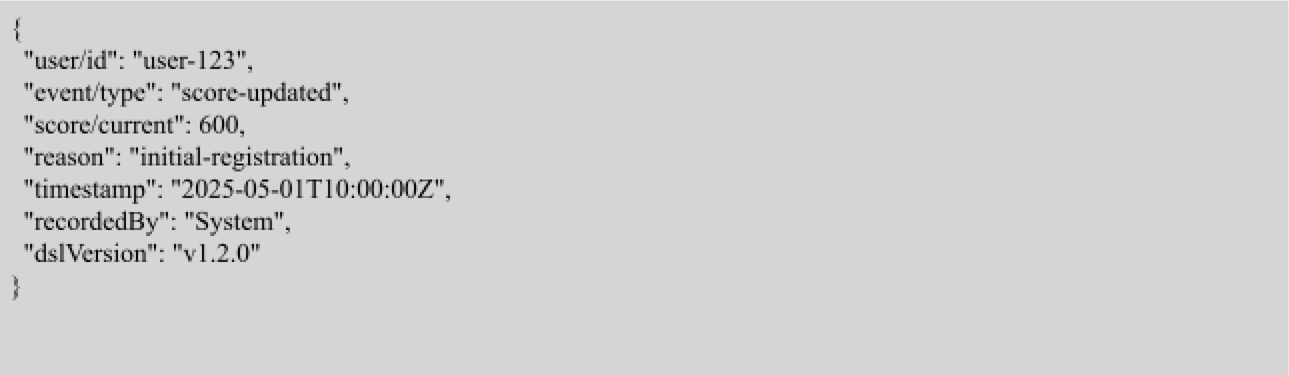
In this system, all data change histories are recorded and persisted as transaction logs in XTDB, ensuring the ability to accurately reproduce, verify, and roll back past states. This enables explicit tracking of "who," "when," and "what"for each processing step, providing high reliability for operational troubleshooting, internal controls, and audit trails.

For public-facing applications, such as local currencies, transparency, legitimacy, and accountability are critical. This system is designed to withstand external verification and audit requests, supporting long-term, trustworthy operations as a foundational infrastructure.

Example: Transaction History and Verification for Local Currency Bonuses

① Initial Score Registration Log (May 1, 2025, 10:00)

At 10:00 AM on May 1, 2025, the system registers an initial score of 600 for a user (ID: user-123). This event is recorded as a "score-updated" event, with the reason field clearly indicating "initial-registration," signifying that this is the initial score setting associated with the user's first registration.



```
{
 "user/id": "user-123",
 "event/type": "score-updated",
 "score/current": 600,
 "reason": "initial-registration",
 "timestamp": "2025-05-01T10:00:00Z",
 "recordedBy": "System",
 "dslVersion": "v1.2.0"
}
```

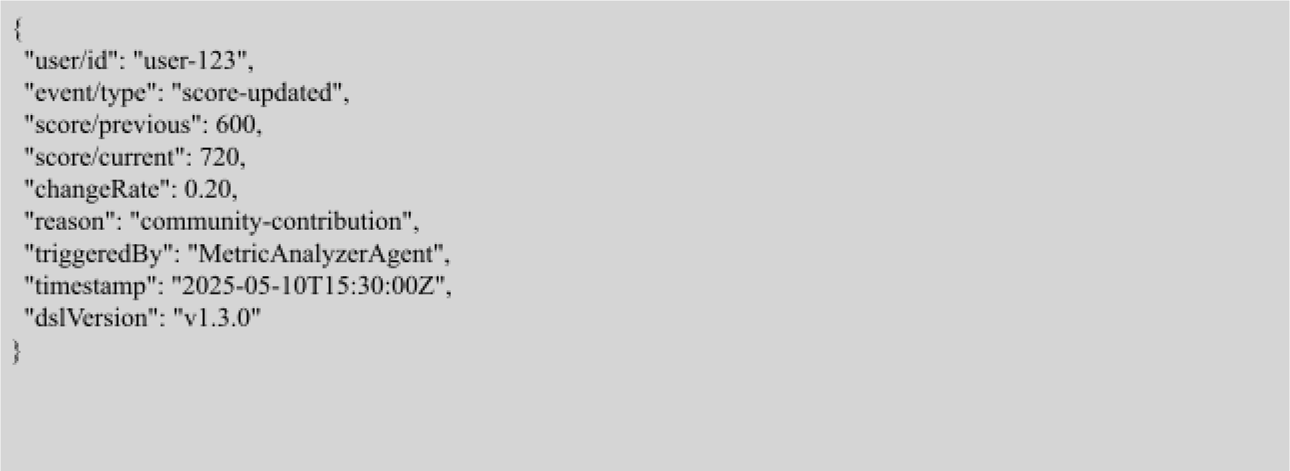
Additionally, this transaction is recorded by the System and uses DSL version v1.2.0, providing a clear trace of the processing logic applied.

② Score Change Due to User Action → Bonus Eligibility via Policy Logic (May 10, 2025)

At 3:30 PM on May 10, 2025, the score for user-123 increased from 600 to 720, reflecting a 20% gain (+120 points).

This change was automatically evaluated by the MetricAnalyzerAgent, with the reason explicitly documented as "community-contribution."

The evaluation process utilized DSL version v1.3.0, meaning the score change logic and underlying rationale are based on the policy definitions from this specific DSL version.



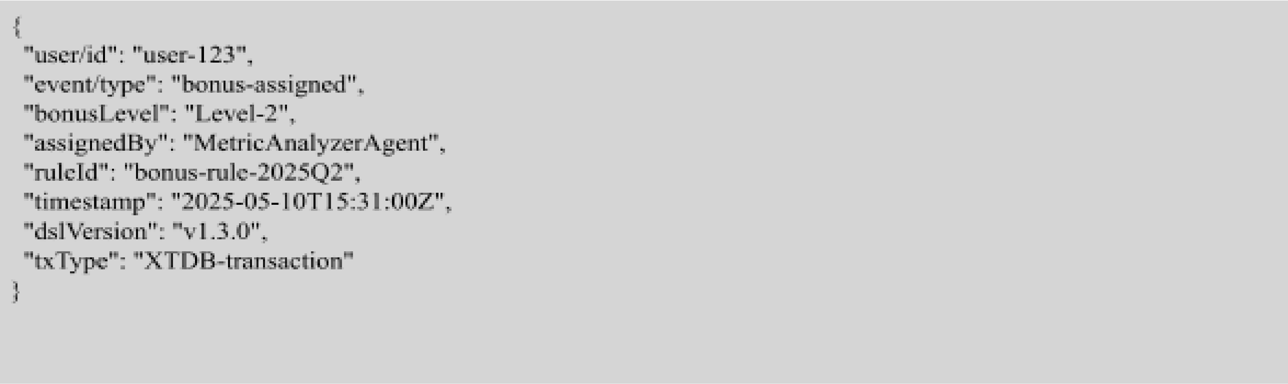
```
{
 "user/id": "user-123",
 "event/type": "score-updated",
 "score/previous": 600,
 "score/current": 720,
 "changeRate": 0.20,
 "reason": "community-contribution",
 "triggeredBy": "MetricAnalyzerAgent",
 "timestamp": "2025-05-10T15:30:00Z",
 "dslVersion": "v1.3.0"
}
```

③ Bonus Level Assignment Log (May 10, 2025)

At 3:31 PM on May 10, 2025, a "Level-2" bonus level was assigned to user-123.

This action was executed by the MetricAnalyzerAgent based on the score increase analysis, with the bonus assignment clearly linked to the policy rule "bonus-rule-2025Q2" as the underlying justification.

The processing used DSL version v1.3.0, meaning the bonus level determination followed the rule definitions specified in this DSL version.



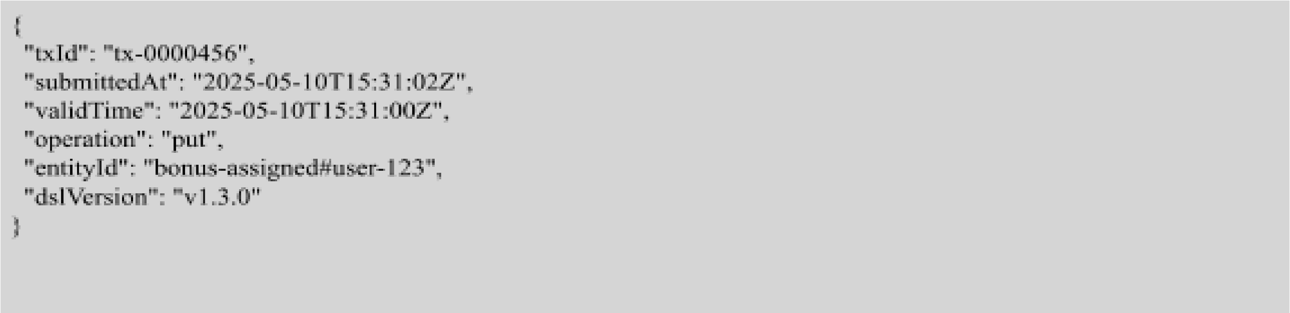
```
{
 "user/id": "user-123",
 "event/type": "bonus-assigned",
 "bonusLevel": "Level-2",
 "assignedBy": "MetricAnalyzerAgent",
 "ruleId": "bonus-rule-2025Q2",
 "timestamp": "2025-05-10T15:31:00Z",
 "dslVersion": "v1.3.0",
 "txType": "XTDB-transaction"
}
```

Additionally, the transaction is logged as an XTDB transaction (txType: XTDB-transaction), ensuring complete traceability within the XTDB database.

④ Transaction Metadata (XTDB Transaction Log)

A transaction with a valid time (logical timestamp) of May 10, 2025, 15:31:00 was submitted to the system at 15:31:02.

This transaction (ID: tx-0000456) is of type "put", indicating that a record with the entity ID "bonus-assigned#user-123" was registered in XTDB.



```
{
 "txId": "tx-0000456",
 "submittedAt": "2025-05-10T15:31:02Z",
 "validTime": "2025-05-10T15:31:00Z",
 "operation": "put",
 "entityId": "bonus-assigned#user-123",
 "dslVersion": "v1.3.0"
}
```

The processing for this transaction was executed in accordance with the schema and logic defined in DSL version v1.3.0, ensuring consistent application of system rules.

Analysis and Verification Using This Transaction Data

This transaction data serves as a core component of the system's data infrastructure, ensuring high transparency and traceability for processes like bonus distribution and contribution evaluation. Possible use cases for this data include:

1. Verification of Proper Bonus Distribution:

- To verify that bonuses were correctly issued, the system can cross-check "bonus-assigned" events with the preceding "score-updated" events in chronological order, confirming that the proper procedures were followed.
- This ensures that each bonus assignment was based on a legitimate sequence of contributions.

2. Validation of Policy Consistency:
    - To confirm that decisions were made in accordance with the latest rules, the dslVersion field in each transaction can be referenced.
    - This guarantees that the evaluation logic used for each score update or bonus assignment aligns with the correct version (DSL v1.3.0), ensuring consistency in policy application.

3. Rollback and Error Correction:
    - In the event of user complaints or processing errors, the system supports transaction-level rollback (undo), allowing specific transactions to be reversed based on their txId.
    - This capability enables targeted corrections without disrupting the integrity of the overall system, supporting flexible recovery measures.

4. External Audit and Accountability:
    - In cases where external accountability is required, such as third-party audits, the system can present a complete chain of events linked to a specific txId.
    - This provides a precise, chronological record of "who", "what", and "when" for each action, demonstrating the system's overall transparency and reliability.

By capturing causal relationships, time sequences, and transaction ownership, the XTDB transaction logs provide a robust foundation for public sector applications, supporting transparency, accountability, reproducibility, and traceability as essential operational features.

(6) Future Scalability and Distributed Processing Support

XTDB is designed to support node-based distributed deployment, allowing for future scale-out and simultaneous multi-site usage.

Additionally, it can be integrated with decentralized, tamper-resistant off-chain storage solutions like IPFS, enabling a more resilient and verifiable data architecture.

With these capabilities, XTDB serves as more than just a traditional database — it acts as a critical infrastructure component that supports:

- Reliable Reconstruction of Historical States
- Proof of Contribution Authenticity
- Loosely Coupled Agent Architectures

This foundational role makes XTDB a cornerstone for building scalable, trustworthy systems that can adapt to evolving operational requirements.

3.2.3. Metric Analyzer Agent

The Metric Analyzer Agent is a core analysis and optimization component within this system, responsible for evaluating user contributions from multiple perspectives, scoring them accordingly, and integrating this data into the reward design process.

(1) Multi-Scoring for Contribution Activities

The Metric Analyzer Agent uses a multi-scoring approach that independently calculates and manages scores across various contribution categories, based on a wide range of user activity logs. Specifically, it generates scores in the following four key areas:

- Activity Score:
    - ◦ Based on physical and time-based actions, such as event participation, task completion, community cleanup, and volunteer work.
- Knowledge Score:
    - ◦ Evaluates intellectual contributions, including FAQ responses, knowledge posts, and the sharing of expert insights.
- Network Score:
    - ◦ Measures social connectivity and relationship-building activities, including mentions, comments, introductions, and overall interaction frequency.
- Economic Score:
    - ◦ Reflects direct and indirect contributions to the local economy, such as community currency usage, product purchases, and the impact of user recommendations.

These scores are dynamically calculated using predefined weighting models (rule-based) or machine learning algorithms like XGBoost and SVM, allowing for flexible adaptation based on campaign goals and target audiences.

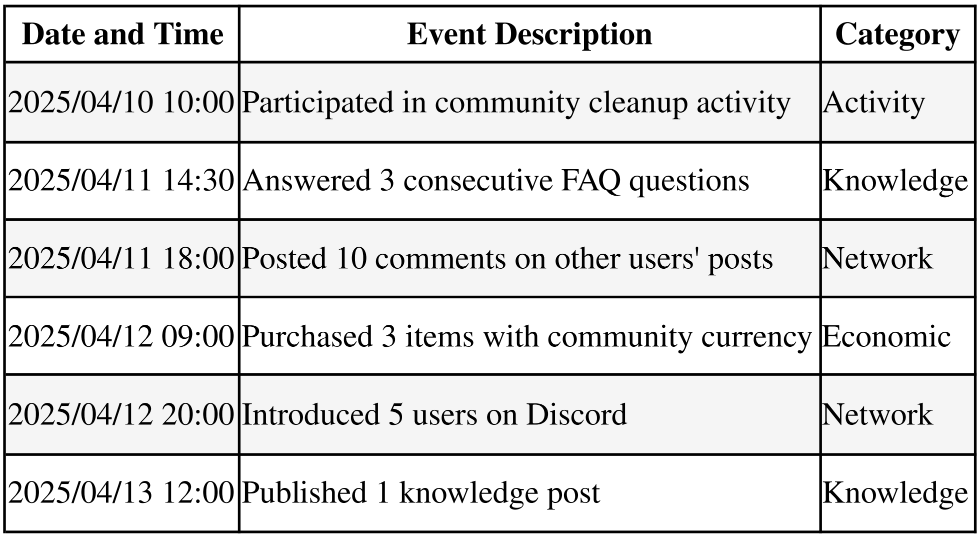
The scoring models can be switched depending on the campaign context, providing a real-time, quantitatively grounded evaluation system for user contributions.

The resulting scores are then integrated with various other agents for reward design (e.g., NFT, tokens), targeting, and dashboard visualization, supporting fair and transparent user evaluation while encouraging active participation.

Example: Scoring Process for a User (user_001)

Input Logs (Structured Event Data Stored in XTDB)

Assume the following structured event logs for a participant recorded between April 10, 2025 and April 13, 2025:



Date and Time	Event Description	Category
2025/04/10 10:00	Participated in community cleanup activity	Activity
2025/04/11 14:30	Answered 3 consecutive FAQ questions	Knowledge
2025/04/11 18:00	Posted 10 comments on other users' posts	Network
2025/04/12 09:00	Purchased 3 items with community currency	Economic
2025/04/12 20:00	Introduced 5 users on Discord	Network
2025/04/13 12:00	Published 1 knowledge post	Knowledge

These logs reflect the various ways in which the citizen has contributed to the local community, as detailed below:

Activity Category:

- Community Cleanup Participation (2025/04/10 10:00)
  - Expected to receive a high score as it directly contributes to the local environment and community well-being.

Knowledge Category:

- Answered 3 Consecutive FAQ Questions (2025/04/11 14:30)
- Published 1 Knowledge Post (2025/04/13 12:00)
  - These actions support knowledge sharing and problem-solving within the community, potentially qualifying the user for TDC (Tokenized Digital Currency) or NFT issuance.

Network Category:

- Posted 10 Comments on Other Users' Posts (2025/04/11 18:00)
- Introduced 5 Users on Discord (2025/04/12 20:00)
  - These activities promote community engagement and strengthen social connections within the network.

Economic Category:

- Purchased 3 Items with Community Currency (2025/04/12 09:00)
  - Directly contributes to the circulation of the local economy through economic transactions.

2. Scoring Logic (Rule-Based with Weighted Evaluation)

In a resident participation promotion project using a local currency in a particular city, citizen contributions are categorized into the following four main categories.

Each category has a corresponding scoring logic, enabling the system to quantitatively assess a wide range of participant behaviors, ensuring fair and transparent evaluations.

The scoring logic for each contribution category in the resident participation promotion project is as follows:

1. Activity Score:
   - Local Event Participation: +30 points per event
2. Knowledge Score:
   - FAQ Answer: +10 points per item
   - Knowledge Post: +20 points per item
3. Network Score:
   - Comment: +2 points per item
   - Introduction: +5 points per item
4. Economic Score:
   - Product Purchase (using local currency): +10 points per item

The details for each score category are as follows:

1. Action Score

- Target Actions: Participation in local events and cleanup activities
- Scoring Logic: +30 points per local event participation
- Example: Participating in one community cleanup activity → +30 points
- Characteristics: High scores for direct community contributions, emphasizing active participation and volunteerism.

2. Knowledge Score

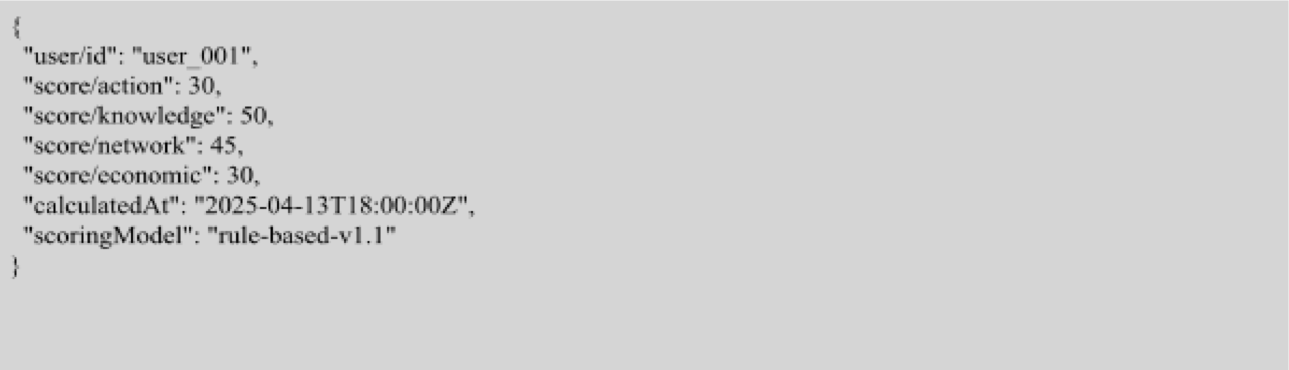
- Target Actions: FAQ responses, knowledge posts
- Scoring Logic:
  - FAQ: +10 points per response
  - Knowledge Post: +20 points per post
- Example: Answering 3 FAQ questions → +30 points (3 × 10 points), posting 1 knowledge article → +20 points
- Characteristics: Rewards contributions that share knowledge and solve problems within the community.

3. Network Score

- Target Actions: Comments on other users' posts, introductions
- Scoring Logic:
  - Comment: +2 points per comment
  - Introduction: +5 points per introduction
- Example: 10 comments → +20 points (10 × 2 points), 5 introductions → +25 points (5 × 5 points)
- Characteristics: Encourages active communication and relationship building within the community.

4. Economic Score

- Target Actions: Purchases made with local currency
- Scoring Logic: +10 points per item purchased
- Example: Purchasing 3 items → +30 points (3 × 10 points)

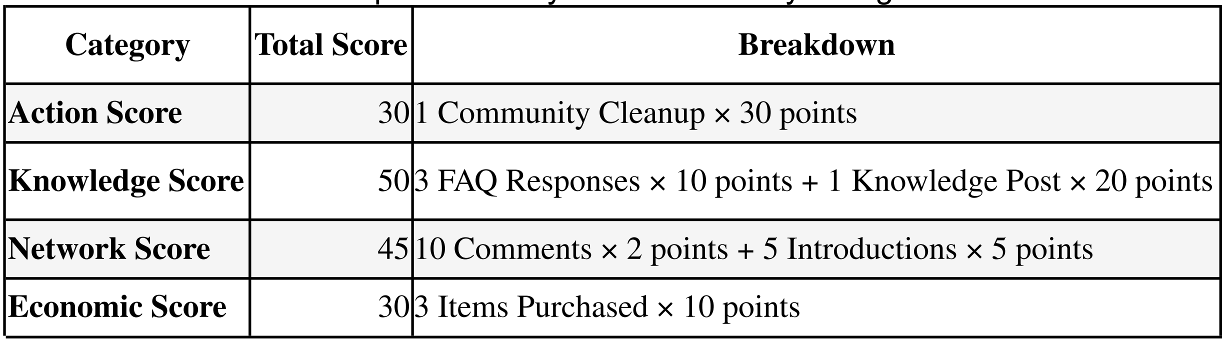


```
{
 "user/id": "user_001",
 "score/action": 30,
 "score/knowledge": 50,
 "score/network": 45,
 "score/economic": 30,
 "calculatedAt": "2025-04-13T18:00:00Z",
 "scoringModel": "rule-based-v1.1"
}
```

- Characteristics: Incentivizes economic activity that supports the local economy.

Evaluating these diverse contributions helps promote balanced participation without favoring any single category, encouraging long-term engagement and a healthy community ecosystem.

3. Score Calculation Results
The score calculations are performed by the Metric Analyzer Agent.



Category	Total Score	Breakdown
Action Score	30	1 Community Cleanup × 30 points
Knowledge Score	50	3 FAQ Responses × 10 points + 1 Knowledge Post × 20 points
Network Score	45	10 Comments × 2 points + 5 Introductions × 5 points
Economic Score	30	3 Items Purchased × 10 points

As a result, the multi-score record stored in XTDB will be structured as follows.
These scores are used in subsequent processes, such as NFT reward thresholds, leaderboard visualization, contribution dashboards, and bonus design.

(2) Historical Management through XTDB Integration

In this system, score change trends are aggregated and analyzed to quantitatively evaluate the impact of specific campaign strategies, such as "how much each contribution category was influenced by a particular campaign" or "how user participation behaviors changed before and after a policy update."

All score calculations for each user are recorded and stored in bitemporal format in XTDB, creating a structured history that allows for consistent analysis of the causal relationships between contributions and reward histories over time.

This enables precise verification and explanation of the "score → reward" process, including:

- When the action occurred

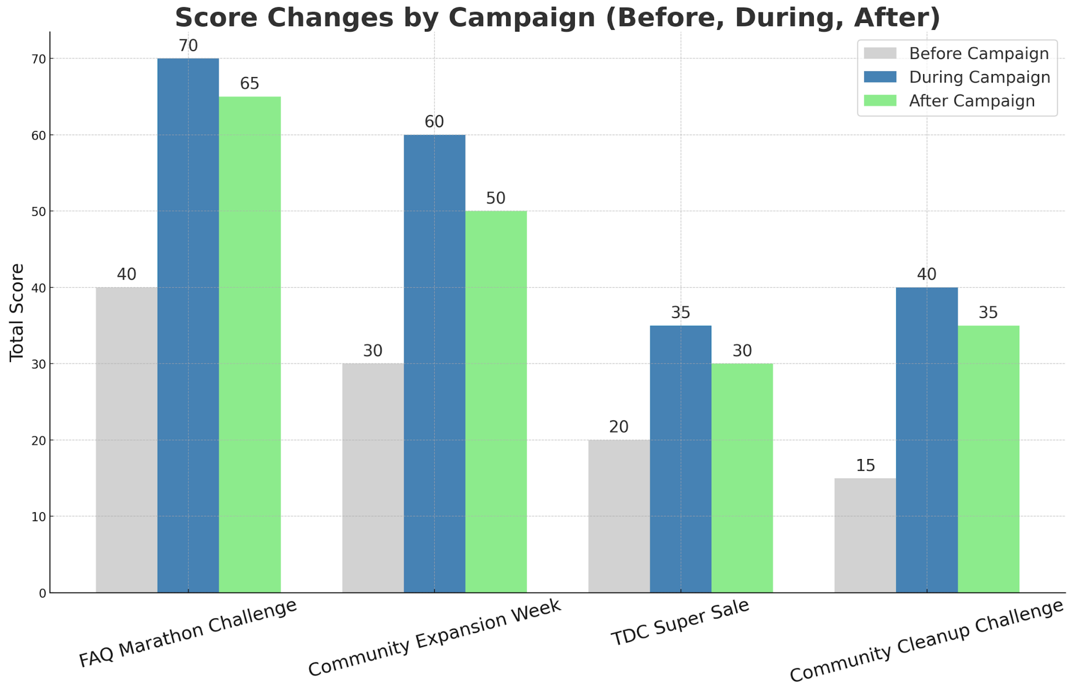
- What kind of action triggered the score change
- How much the score increased
- When the score change resulted in a reward (e.g., NFT or token)

This history integration capability transforms the system into more than just a scoring engine. It acts as a comprehensive community contribution analysis infrastructure, supporting accountability, campaign evaluation, and transparency.

Example of Score Change Analysis

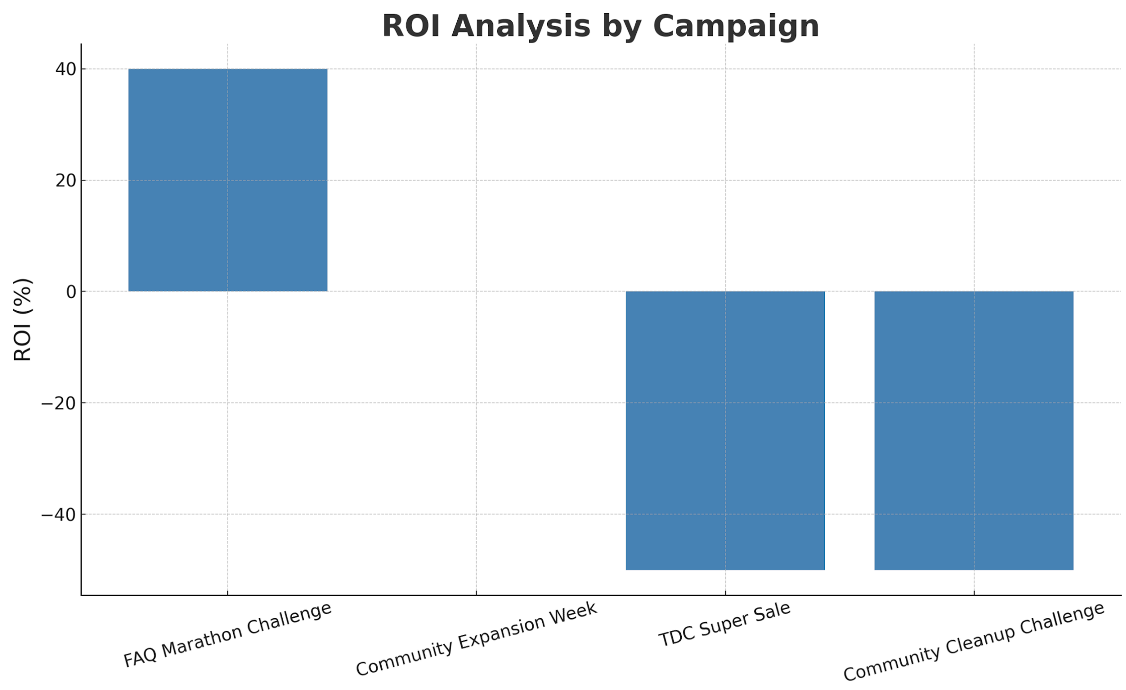
To illustrate this, here is an example of score changes during the following campaigns:

- FAQ Marathon Challenge
- Community Expansion Week
- TDC Super Sale
- Community Cleanup Challenge



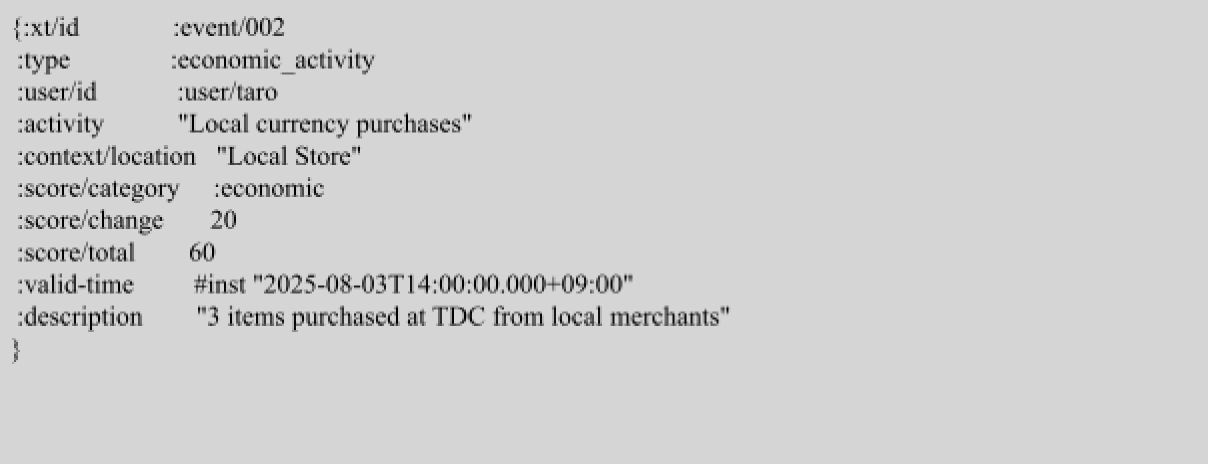
The Community Cleanup Challenge showed the most significant impact, with a +166% increase, indicating the strongest response to direct, hands-on activities. Additionally, it is notable that scores remained high even after the end of each campaign, reflecting sustained engagement.

Evaluating the ROI (Return on Investment) for each campaign reveals the following:



Based on this data, combining high-impact campaigns like Community Cleanup Challenge and TDC Super Sale can potentially enhance economic effects. For example, a campaign like "Eco Purchase Multiplier" (where users who participate in cleanup activities receive additional bonuses for each purchase made with TDC during the campaign period) could be effective.

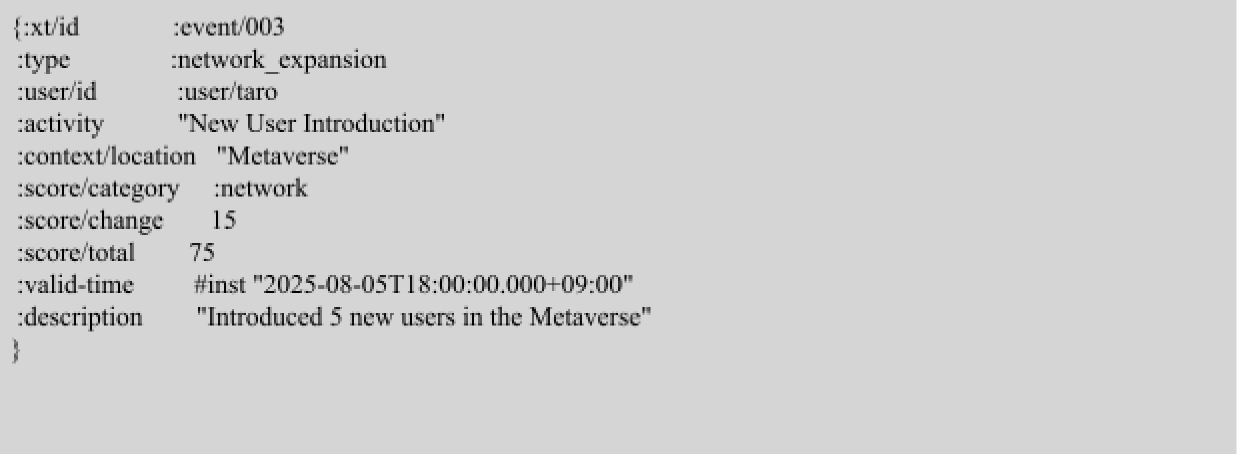
Alternatively, combining campaigns with strong ROI, such as the FAQ Marathon Challenge and TDC Super Sale, could improve overall ROI. For instance, a "Knowledge Purchase Multiplier" campaign (offering additional bonuses for TDC purchases made by users who actively respond to FAQs or share knowledge) would encourage both economic activity and knowledge contribution.



```
{:xt/id            :event/002
 :type             :economic_activity
 :user/id           :user/taro
 :activity          "Local currency purchases"
 :context/location   "Local Store"
 :score/category     :economic
 :score/change       20
 :score/total        60
 :valid-time         #inst "2025-08-03T14:00:00.000+09:00"
 :description        "3 items purchased at TDC from local merchants"
}
```

Additionally, data like the following can be recorded in XTDB through the Metric Analyzer Agent, ensuring that these contribution actions are properly logged and evaluated.

If a user has a history of economic_activity, it may be possible to assign them cashback programs or region-specific discount coupons based on their local currency usage frequency.



```
{:xt/id            :event/003
 :type             :network_expansion
 :user/id           :user/taro
 :activity          "New User Introduction"
 :context/location   "Metaverse"
 :score/category     :network
 :score/change       15
 :score/total       75
 :valid-time        #inst "2025-08-05T18:00:00.000+09:00"
 :description       "Introduced 5 new users in the Metaverse"
}
```

For example, using data like the following:

If a user has a history of network_expansion, it may be possible to implement a referral ranking system (e.g., Bronze, Silver, Gold) that awards special badges based on their rank.

Additionally, offering extra points for SNS shares can further enhance their influence and increase the overall reach of the network.

(3) Integrated Reward Optimization Function

This system not only outputs user contribution scores but also includes a reward optimization function that dynamically determines and executes the most suitable rewards based on the user's score details.

Specifically, it calculates the optimal combination of rewards (e.g., local currency, NFTs, discount tokens, status upgrades) in real time, based on the user's activity history, contribution patterns, and current score level.

This optimization process is not based on static, single-metric rules but is designed as a context-aware reward strategythat considers:

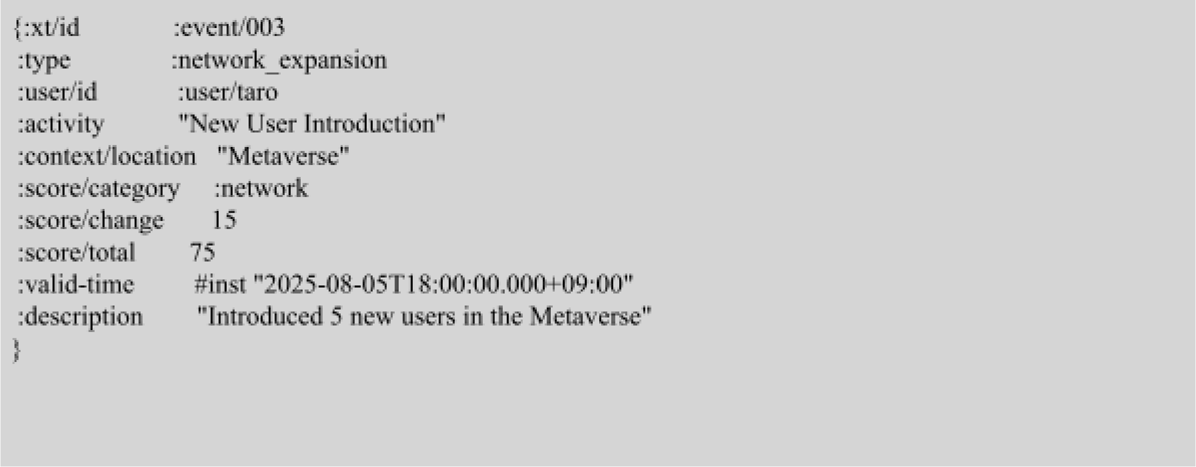
- Participation Patterns (e.g., frequency, contribution types, past reward history)
- Regional Campaign Objectives (e.g., first-time participation, ongoing engagement, local event promotion)

Once the optimal reward combination is determined (e.g., NFT + 20 TDC + 10% discount coupon), it is generated as a structured reward script by the LLM Agent and sent to the Cardano Integration Agent.

The Cardano Agent then automatically handles the token or NFT issuance, wallet transfers, and blockchain transaction recording based on this instruction.

Example:

Suppose the following data is stored in XTDB:

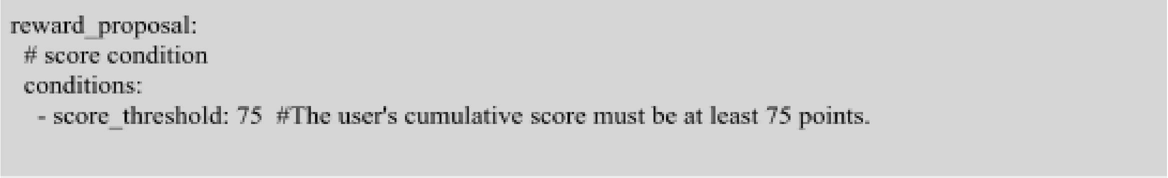


```
{:xt/id            :event/003
 :type             :network_expansion
 :user/id           :user/taro
 :activity          "New User Introduction"
 :context/location   "Metaverse"
 :score/category     :network
 :score/change       15
 :score/total       75
 :valid-time        #inst "2025-08-05T18:00:00.000+09:00"
 :description       "Introduced 5 new users in the Metaverse"
}
```

"User Taro introduced 5 new users in the Metaverse at 6:00 PM on August 5, 2025, resulting in a 15-point increase in network score, bringing their total cumulative score to 75 points."

Based on the reward proposal:

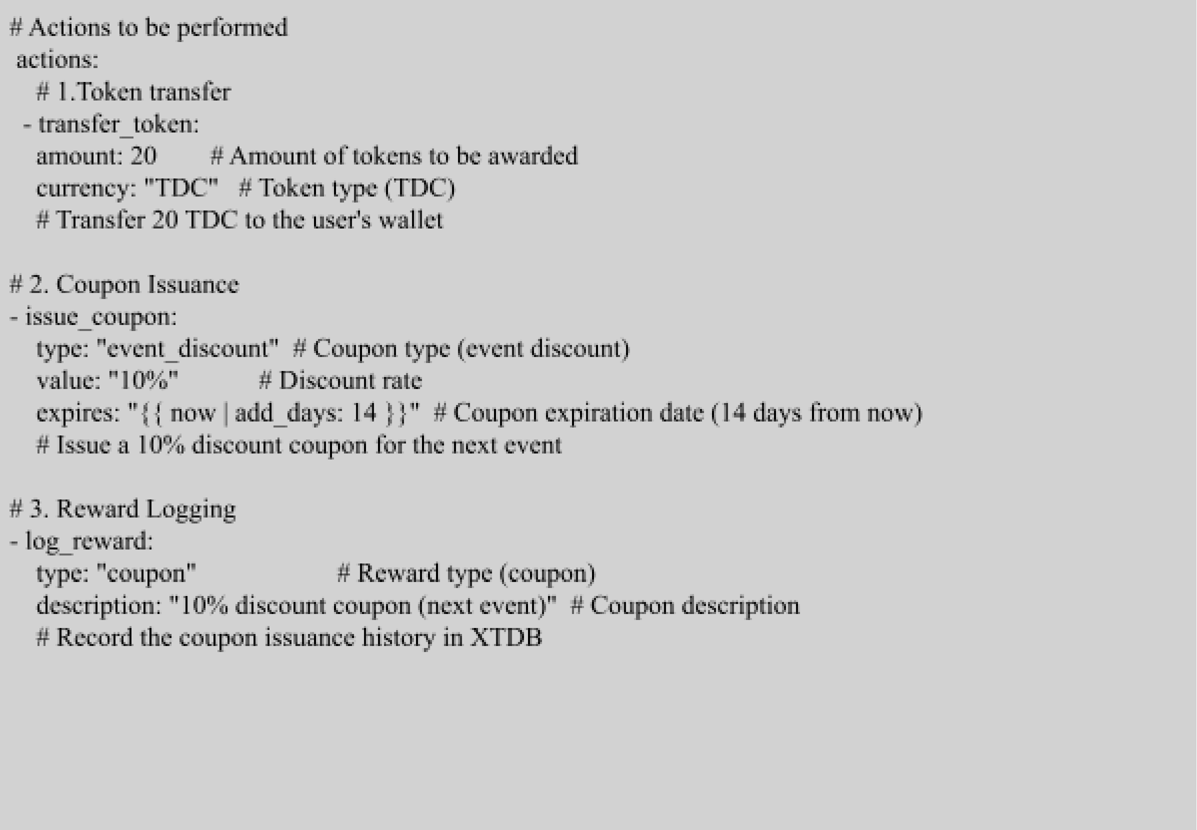
"Users with a score of 75 points or more receive 20 TDC tokens and a 10% discount coupon for the next event. Additionally, duplicate issuance of existing NFTs should be avoided."



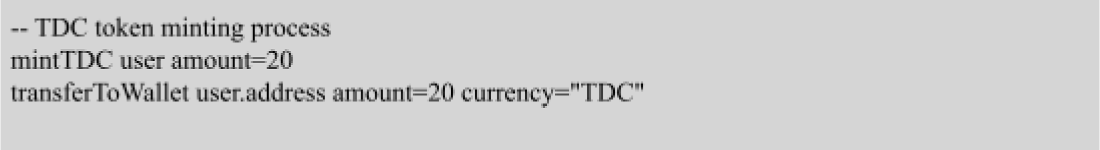
```
reward_proposal:
  # score condition
  conditions:
    - score_threshold: 75  #The user's cumulative score must be at least 75 points.
```

the following DSL is generated by the **LLM Agent**:

The following three processes are performed using the Cardano agent described below.



```
# Actions to be performed
 actions:
   # 1.Token transfer
  - transfer_token:
   amount: 20        # Amount of tokens to be awarded
   currency: "TDC"   # Token type (TDC)
   # Transfer 20 TDC to the user's wallet

# 2. Coupon Issuance
- issue_coupon:
   type: "event_discount"  # Coupon type (event discount)
   value: "10%"            # Discount rate
   expires: "{{ now | add_days: 14 }}"  # Coupon expiration date (14 days from now)
   # Issue a 10% discount coupon for the next event

# 3. Reward Logging
- log_reward:
   type: "coupon"                    # Reward type (coupon)
   description: "10% discount coupon (next event)"  # Coupon description
   # Record the coupon issuance history in XTDB
```

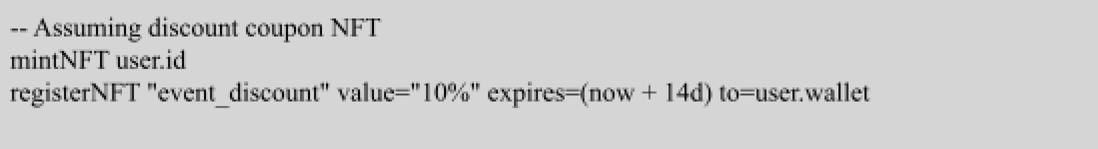


```
-- TDC token minting process
mintTDC user amount=20
transferToWallet user.address amount=20 currency="TDC"
```

Minting and Transferring TDC Tokens

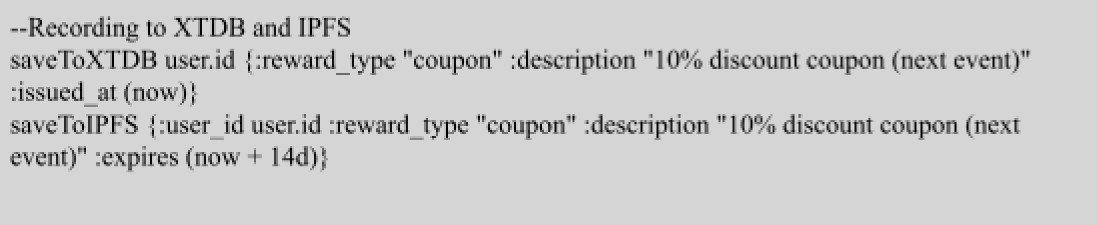
- **mintTDC:** Mints 20 TDC tokens
- **transferToWallet:** Transfers the tokens to the user's wallet

Issuing Discount Coupon NFTs



```
-- Assuming discount coupon NFT
mintNFT user.id
registerNFT "event_discount" value="10%" expires=(now + 14d) to=user.wallet
```

- **mintNFT:** Issues a discount coupon NFT
- **registerNFT:** Registers the NFT with the event-linked DApp

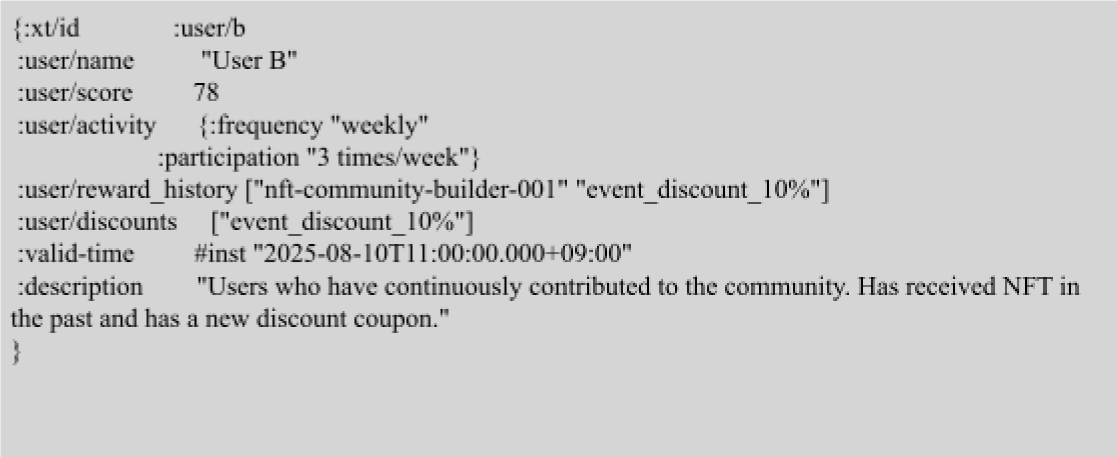


```
--Recording to XTDB and IPFS
saveToXTDB user.id {:reward_type "coupon" :description "10% discount coupon (next event)"
:issued_at (now)}
saveToIPFS {:user_id user.id :reward_type "coupon" :description "10% discount coupon (next
event)" :expires (now + 14d)}
```

Saving Transaction Logs

- **saveToXTDB:** Records the transaction history in XTDB
- **saveToIPFS:** Stores a backup of the transaction history on IPFS

Resulting User Data in XTDB



```
{:xt/id             :user/b
 :user/name          "User B"
 :user/score        78
 :user/activity      {:frequency "weekly"
                    :participation "3 times/week"}
 :user/reward_history ["nft-community-builder-001" "event_discount_10%"]
 :user/discounts     ["event_discount_10%"]
 :valid-time         #inst "2025-08-10T11:00:00.000+09:00"
 :description        "Users who have continuously contributed to the community. Has received NFT in
the past and has a new discount coupon."
}
```

After completing these actions, the user's data is stored in **XTDB** as follows:

- user/reward_history: The newly issued discount coupon is added
- user/discounts: The currently active discount coupon is also included

These actions, defined through DSL, effectively execute the following sequence:

"Mint 20 TDC tokens, transfer them to the user's wallet, issue a discount coupon NFT, register it with the event-linked DApp, and save all records to XTDB and IPFS, then reflect the updates in the View Agent."

This automated reward design mechanism, which adjusts based on each user's situation and regional context, enables maximizing incentive effectiveness while reducing operational costs.

(4) Integration of Cross-Reality Data

This system incorporates Cross-Reality Data through a Cross-Reality Agent to comprehensively evaluate user contributions across both physical and virtual spaces. This includes capturing and utilizing behavior logs from Metaverse, Discord, and other digital platforms to ensure fair and consistent evaluation of user activities.

Key Data Sources Include:

- Metaverse Activities: Event participation, in-world conversations, facilitation, time spent in virtual spaces, interactions with others, and guidance activities.
- Discord Activities: Chat contributions, question responses, thread creation, and empathy reactions (e.g., likes, thanks).

These actions are recorded and analyzed in a time-series format, allowing the Metric Analyzer Agent to integrate them into its scoring processes.

This approach enables the system to fairly evaluate users with diverse participation patterns, such as:

- Users who actively support new participants in the Metaverse, even if they cannot attend physical events.
- Users who share expert knowledge in virtual conversations while continuing to engage offline.

This unified scoring model ensures that contributions in both real and virtual worlds are integrated and fairly represented, supporting holistic and context-aware evaluations.

The resulting integrated scores are then used for reward optimization, role recognition, and incentive design, creating a sustainable contribution evaluation framework suited for the cross-reality era.

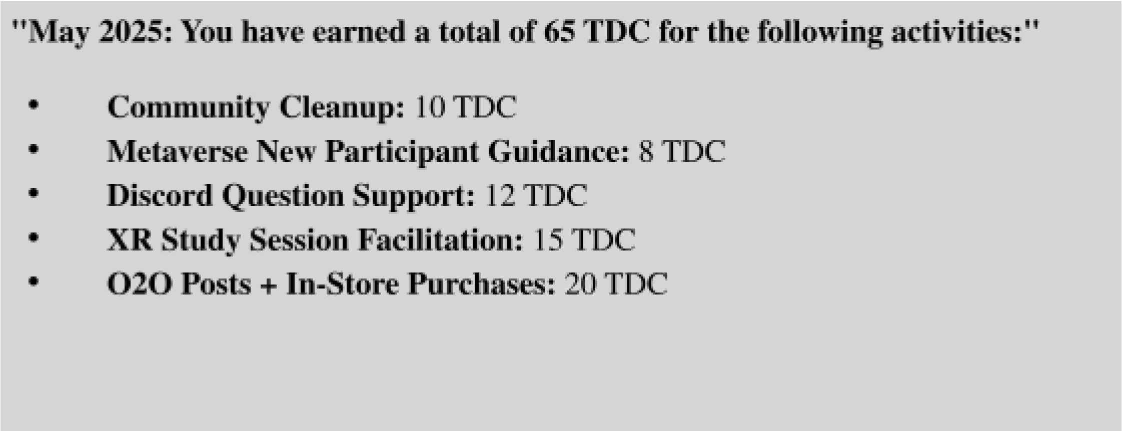
Example of Cross-Reality Data Integration:

Consider a city that operates both real-world events (e.g., shopping street festivals, community cleanups) and virtual space initiatives (e.g., XR meetups, virtual guides, resident meetings) as part of a local revitalization effort.

Participants in these activities are evaluated fairly across both physical and virtual contexts, and receive incentives in the form of the local currency "TDC" based on their contributions.

Suppose the following cross-reality data was collected in both the physical and virtual worlds:

Activity Type	Space	Example of Log Collection	Score Category	Local Currency Example
Community Cleanup Participation	Real	NFC check-in, photo records	Action Score (On-site Participation)	10 TDC
New Participant Guidance in Metaverse	Metaverse	Proximity logs, voice logs, duration of interaction	Support Score (Spatial Contribution)	8 TDC
Answering Questions on Discord	Online	Speech logs, thread references	Knowledge Score (Expertise)	12 TDC
Facilitating Online Study Sessions	Metaverse + Discord	Simultaneous attendance, session logs	Facilitation Score	15 TDC
In-Store Purchases with Linked Posts	Real + Discord	Store QR code, timestamped posts	O2O (Online to Offline) Contribution	20 TDC



"May 2025: You have earned a total of 65 TDC for the following activities:"

- **Community Cleanup:** 10 TDC
- **Metaverse New Participant Guidance:** 8 TDC
- **Discord Question Support:** 12 TDC
- **XR Study Session Facilitation:** 15 TDC
- **O2O Posts + In-Store Purchases:** 20 TDC

Based on this user activity, the Metric Analyzer Agent calculates the integrated scores for each user. The Cardano Agent then executes token issuance and wallet transfers, followed by a message sent through the View Agent as follows:

In this way, the integrated evaluation of real and virtual activities, combined with the fair and cross-context issuance of local currency, serves as a foundational infrastructure for next-generation regional economic ecosystems that can accommodate diverse participation styles beyond physical constraints.

(5) Score Visualization and Feedback Support

In this system, user contributions are not only scored but also visualized through a user-friendly UI in collaboration with the View Agent. This approach allows users to intuitively understand the components of their scores and the reasons behind score changes, enhancing their sense of satisfaction and motivation.

Key Features of the Visualization UI:

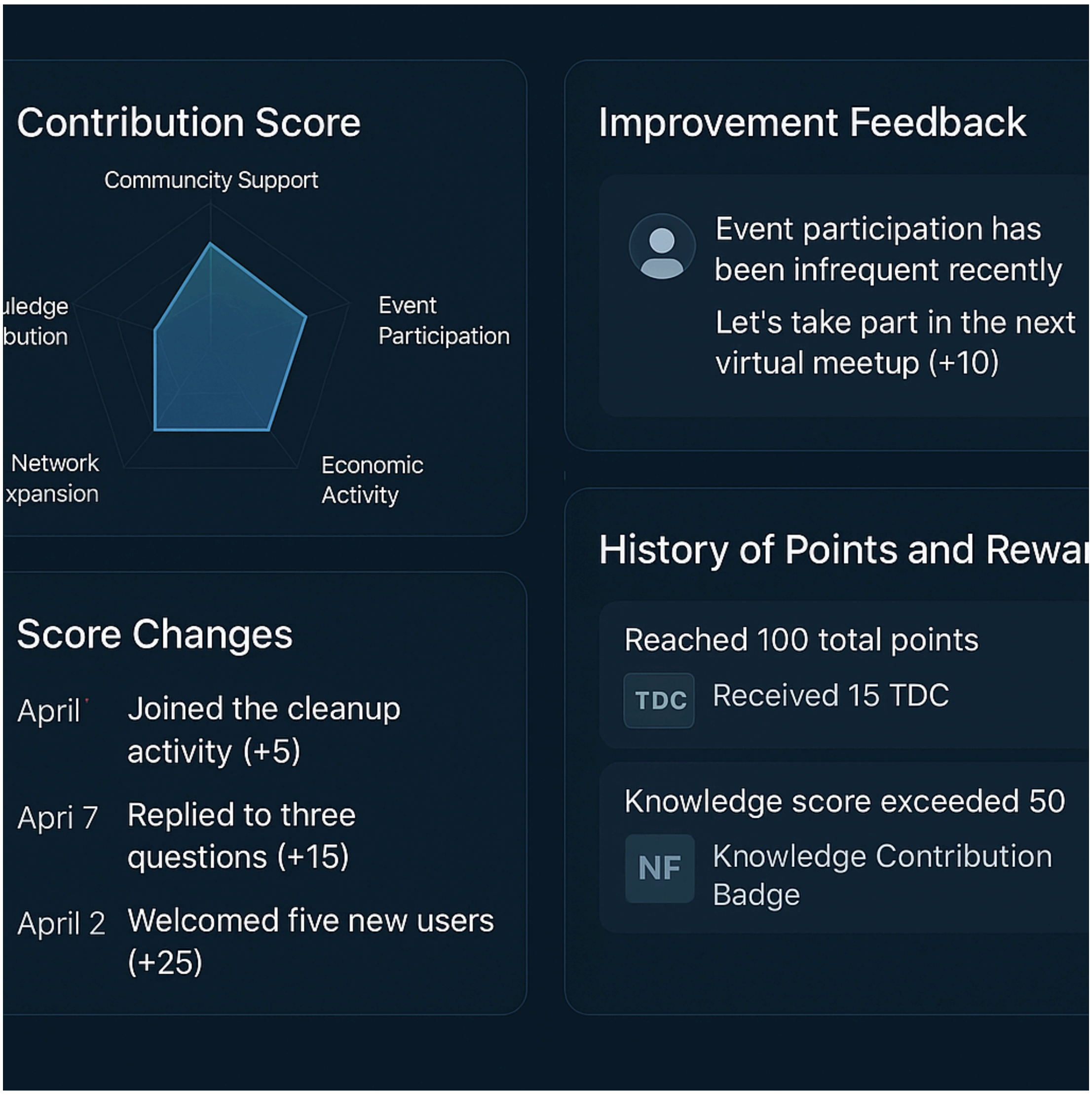
- Score Breakdown:
    - Users can see the detailed composition of their scores (e.g., knowledge sharing, support activities, event participation) through visual formats like radar charts and timelines.
- Action-Triggered Score Insights:
    - When a score increases, the UI highlights specific actions that contributed to the score change, such as "Event Presentation on [Date]" or "5 Replies to New Users", providing clear context for the user's achievements.
- Personalized Feedback:
    - In addition, the LLM Agent generates personalized improvement advice, such as "Your support activities have been lower recently. Try engaging more in question threads."
    - This natural language feedback encourages proactive participation and continuous contribution.

Benefits:

- For users, this approach enables them to independently choose their next actions, fostering long-term engagement and personal growth.
- For administrators, it provides a clearer understanding of each user's contribution trends and areas for improvement, leading to more precise reward design and role assignment decisions.

Example:

Below is an example of a Cross-Reality Urban Development Project UI for visualizing local currency contributions, providing real-time insights into community engagement.



Improvement Feedback

Event participation has been infrequent recently

Let's take part in the next virtual meetup (+10)

Score Changes

April	Joined the cleanup activity (+5)
Apri 7	Replied to three questions (+15)
April 2	Welcomed five new users (+25)

History of Points and Rewa

Reached 100 total points

TDC Received 15 TDC

Knowledge score exceeded 50

NF Knowledge Contribution Badge

This UI is designed to link local currency rewards with user scores based on both real and virtual activities. By visualizing scores and providing targeted feedback, it aims to foster sustained participation and proactive contributions.

Key Features

1. Contribution Score Radar Chart

   ◦ Displays multiple contribution categories, such as knowledge sharing, support activities, and local event participation, in a pentagon-shaped radar chart.

   ◦ Each axis represents a specific contribution area, allowing users to quickly assess their overall balance and strengths.

2. Timeline-Linked Score Change Visualization

  - Shows the timing of score increases and the specific actions that triggered them.
  - For example, "August 5 - Introduced 5 new users +15 points", clearly linking activities to score changes for intuitive understanding.

1. Improvement Feedback via LLM
   - Provides personalized improvement suggestions for categories with declining scores, using natural language generated by the LLM Agent.
   - Example: "Your support activities are low. Consider engaging more actively in the next event."

4. Contribution-to-Reward History
   - Visualizes the entire flow from actions to scores, and from scores to rewards (e.g., TDC or NFT), allowing users to easily review their past contributions and the rewards they earned.
   - Makes it clear how past actions directly translated into tangible rewards.

Impact

This type of UI is not just a score display, but a critical element for data-driven behavior improvement, helping users understand their impact and motivating continued engagement.

(6) Score Applications and External Integration

The user scores generated by this system are not just internal indicators for reward calculations, but are also intended for broader applications such as policy planning, target selection, and campaign design by local governments and regional partners.

For example, these scores can be used as tools to quantitatively identify suitable participants for citizen collaboration projects or to offer special benefits to "support-oriented residents" in community volunteer programs.

Flexible, Multi-Dimensional Scoring
The scoring framework is designed to be flexible, incorporating multiple evaluation axes such as knowledge sharing, empathy, support activities, and sustained engagement. This structure allows the scores to serve as generalized indicators of user contributions, independent of any single platform.

As a result, the system can be extended to integrate with other regional currency platforms, external education and training programs, and employment support systems in the future. This cross-regional integration could enable users to share their "local contribution history" and "reputation for collaboration" across different ecosystems.

Potential Use Cases:

1. Inter-Regional Trust Systems:
    - For example, users might be treated as highly trusted participants based on their previous score history when joining regional currency programs in other municipalities.
2. Educational and Professional Applications:
    - Educational institutions could use local contribution scores as criteria for scholarship recommendations, recognizing students who have demonstrated significant community involvement.

Cross-Regional Integration Model Example

In a cross-regional collaboration project involving multiple municipalities, a shared infrastructure based on local currency could enable broader utilization of user contribution scores. This approach could connect various municipalities, educational institutions, and organizations, creating a cross-regional trust score system.

Example Use Cases for Government and External Organizations:

- City A - Collaborative Policy Department
    - Score Usage: Select residents with TDC scores of 80 or higher as participants in collaborative urban development projects
    - Local Currency Connection: Provide additional TDC rewards and project compensation for selected participants
- City B - Social Welfare Council
    - Score Usage: Recommend volunteer opportunities for residents with high empathy and support scores
    - Local Currency Connection: Offer TDC-based compensation and travel support based on activity level
- City C - Chamber of Commerce
    - Score Usage: Distribute TDC discount coupons to students with high sustainability scores for use in local shops
    - Local Currency Connection: Promote local economic circulation and encourage youth retention

- City D - Partner Universities
    - Score Usage: Use contribution scores as part of the criteria for scholarship recommendations
    - Local Currency Connection: Visualize active local engagement and link it to regional education support programs
- City E - New Participating Region
    - Score Usage: Reference past TDC score history to assign initial trust ranks to new users
    - Local Currency Connection: Introduce a "Trust Inheritance System" that allows high-rank rewards and early NFT issuance from the start

Future Expansion Possibilities:

- Score Portability:
    - Users can carry their trust scores across multiple regions, ensuring that their past contributions are recognized even if they move or join a new community. This approach promotes active participation across different regions and strengthens inter-regional collaboration.
- Regional Currency Bridging:
    - Establish mutual trust systems between TDC and other local currencies, enabling seamless economic interactions across regions. This would standardize reward qualifications and benefits, fostering broader community networks and regional revitalization.
- Trust-Based Economies:
    - Scores could be tokenized as NFTs or digital credentials, making them usable in external Web3 services and decentralized social systems. This approach extends the value of user contribution histories, enhancing trust across a wide range of digital ecosystems.

In this way, TDC scores can evolve beyond simple local points, becoming a form of decentralized social capital that reflects trust, behavior, and long-term commitment. This positions the Metric Analyzer Agent not just as a scoring tool, but as an intelligent evaluation and reward agent that supports contribution quantification, personalized reward optimization, and policy integration for next-generation regional economies.

3.2.4. Cardano Blockchain Integration Agent

The Cardano Blockchain Integration Agent is a blockchain processing module responsible for the issuance, distribution, and recording of tokens/NFTs based on user contributions. In this system, it works in collaboration with the LLM Agent (Large Language Model) to automate smart contract generation and transaction construction, enabling flexible, context-aware blockchain operations without human intervention.

(1) Smart Contract Generation Support (LLM × DSL)

In this system, the LLM Agent automatically generates smart contract templates in DSL (Domain-Specific Language)format, based on the score data and user attribute information (e.g., contribution category, influence rank, behavior patterns) provided by the Metric Analyzer Agent.

This DSL acts as an intermediate, human-readable language for defining reward designs and distribution conditionswithin the community, allowing non-engineers to easily verify and customize the logic.

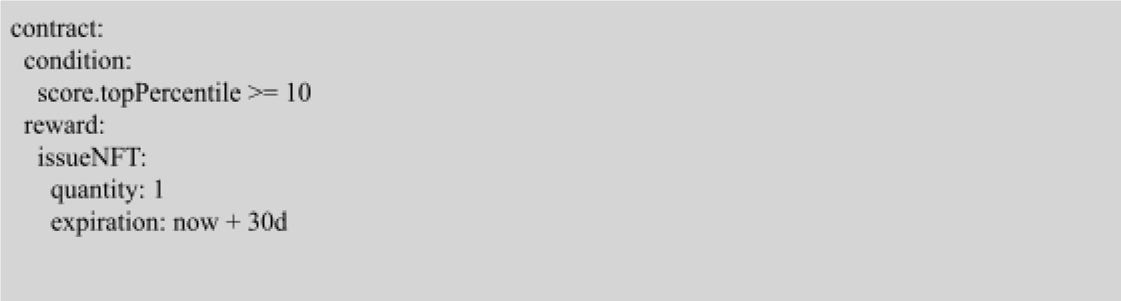
Once generated, the DSL is interpreted and transformed by the Cardano Agent into Plutus-compatible smart contract code, which is then compiled and automatically deployed to the Cardano blockchain.

This end-to-end process enables real-time and structured execution of token/NFT rewards based on user actions and scores.

Example:

Consider the following scenario:

"Issue 1 NFT to the top 10% of users based on score, with a usage expiration of 30 days."



```
contract:
  condition:
    score.topPercentile >= 10
  reward:
    issueNFT:
      quantity: 1
      expiration: now + 30d
```

The corresponding DSL for this smart contract might look like the following:

このDSLは、ユーザーのスコアに基づいてNFTを発行するスマートコントラクトの基本構造を示しており、” condition: score.topPercentile >= 10”でユーザーのスコアが上位10%以内である場合にのみ、この契約が発動し、”reward: issueNFT {quantity: 1 expiration: now + 30d}”で条件を満たしたユーザーに対して、NFT（Non-Fungible Token）を1つ発行することを定義している。

このDSLを用いてCardanoエージェントが生成するhaskelでのPlutusコードは以下のようになる。

The Cardano Agent automatically adds standard checks and tests that are not explicitly defined in the DSL, ensuring robust and secure contract execution.

This mechanism tightly integrates dynamic community evaluation metrics with token economics, enabling a highly transparent, flexible, and autonomous reward design.

Additionally, all generated DSL scripts and executed contract histories are recorded in XTDB, ensuring full traceability and reproducibility for future audits and evaluations.

(2) Conversion of Natural Language Descriptions into Transaction Scripts

In this system, the LLM Agent is capable of interpreting natural language reward rules and condition descriptions entered by users or administrators (e.g., "Distribute 10 ADA to users with a contribution score over 50 this week.") and automatically converting them into structured transaction scripts (in DSL or intermediate language formats).

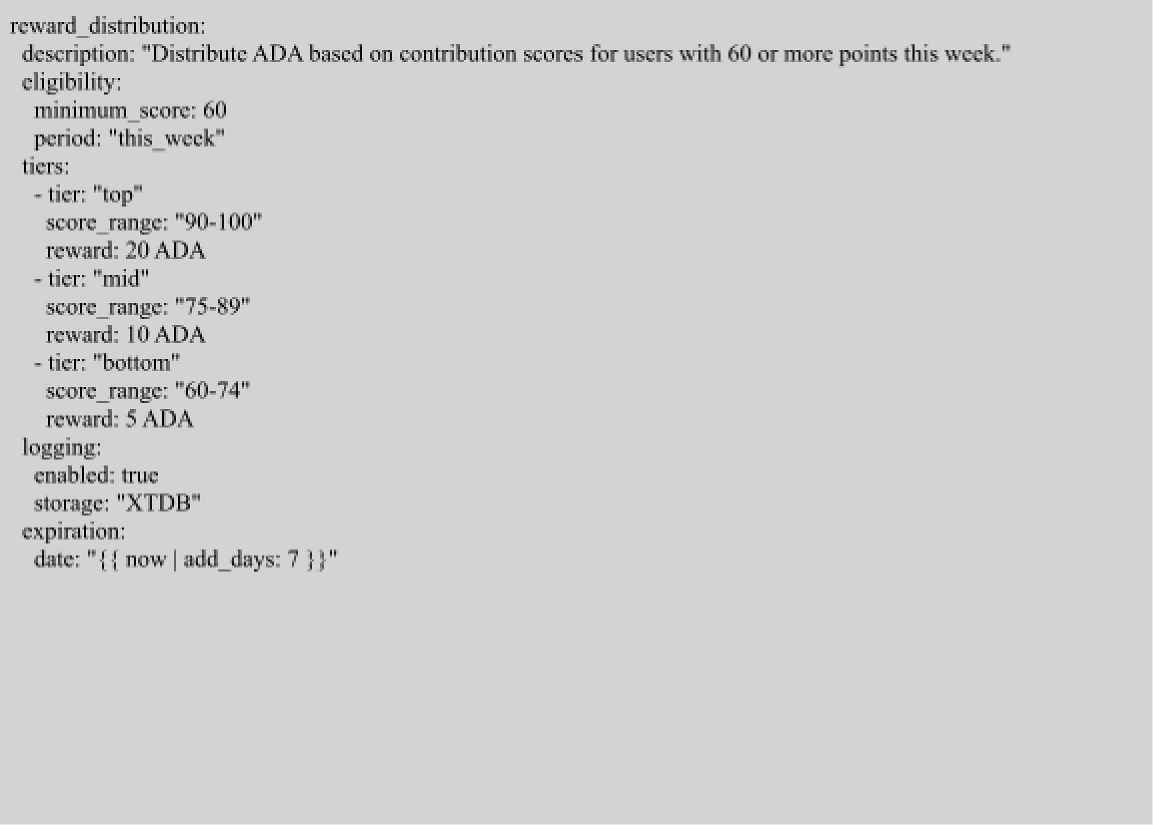
This conversion process eliminates the ambiguity of natural language while maintaining the intuitive readability of the original rule descriptions, resulting in clear, executable logic for smart contract execution.

End-to-End Process:

1. Natural Language Input → DSL Generation:
   - The LLM Agent extracts the precise meaning from natural language inputs and converts them into structured DSL scripts.
2. DSL Verification and Contract Compilation:
   - The generated DSL is then verified by the Cardano Agent, converted into Plutus-compatible smart contract transactions, signed, and recorded on the Cardano blockchain.
3. Full Automation:
   - This end-to-end automation allows the entire process, from rule design to transaction issuance, to be completed using natural language inputs alone.

Example of Natural Language Input:

"Distribute ADA in three tiers based on contribution scores for users with 60 or more points this week: top tier 20 ADA, mid tier 10 ADA, bottom tier 5 ADA."



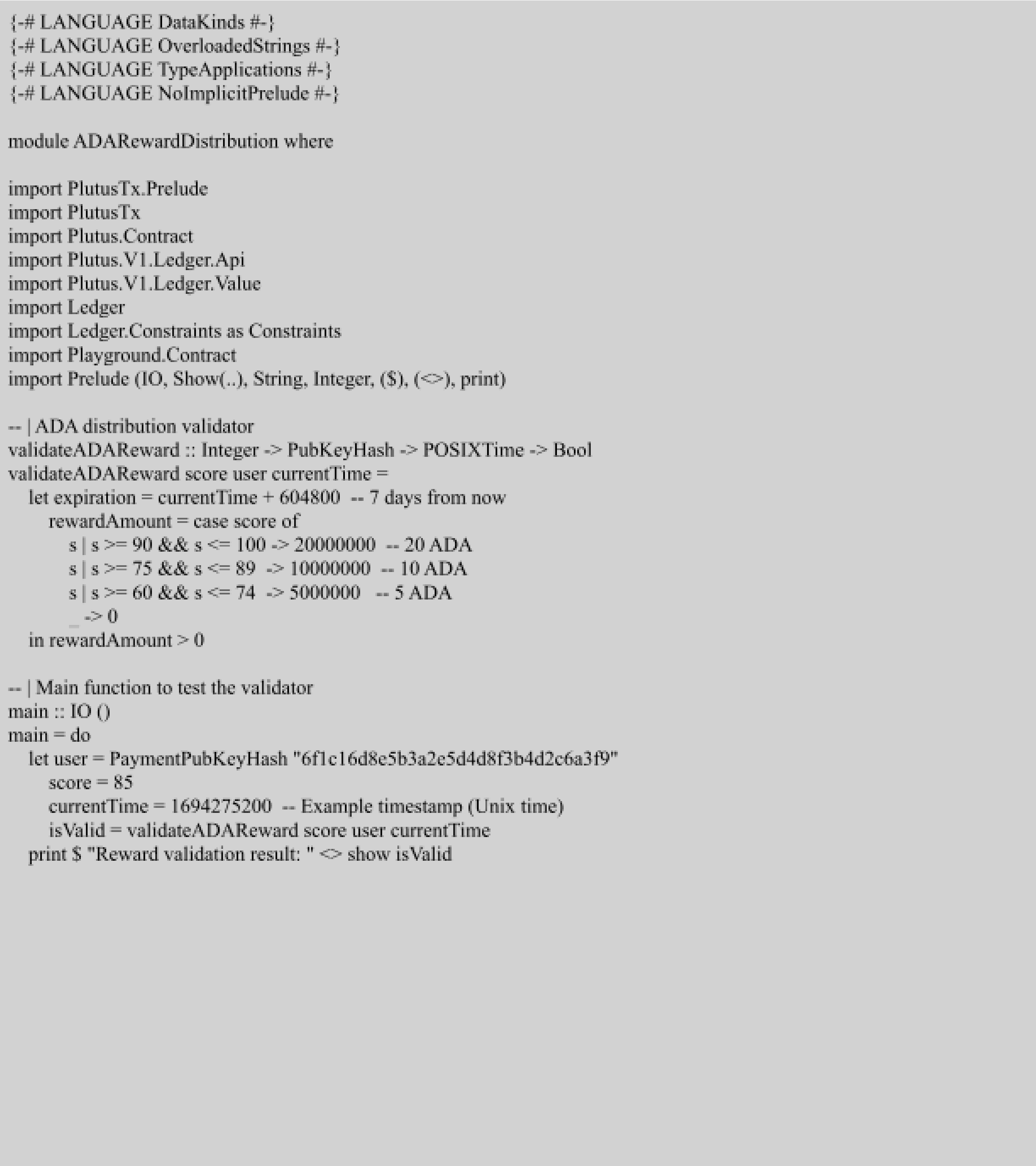
```
reward_distribution:
  description: "Distribute ADA based on contribution scores for users with 60 or more points this week."
  eligibility:
    minimum_score: 60
    period: "this_week"
  tiers:
    - tier: "top"
      score_range: "90-100"
      reward: 20 ADA
    - tier: "mid"
      score_range: "75-89"
      reward: 10 ADA
    - tier: "bottom"
      score_range: "60-74"
      reward: 5 ADA
  logging:
    enabled: true
    storage: "XTDB"
  expiration:
    date: "{{ now | add_days: 7 }}"
```

When processed by the LLM Agent, this input is automatically converted into the following DSL:

This DSL defines the process for distributing ADA (Cardano's native cryptocurrency) based on each user's score, making it ideal for gamification and incentive programs.

The Cardano Agent converts this DSL into the following Plutus transaction, enabling automated token transfers on the Cardano blockchain:

In this way, even complex transactions can be converted from natural language to transaction scripts, allowing non-engineers to intuitively define and execute reward distribution rules.



```
{-# LANGUAGE DataKinds #-}
{-# LANGUAGE OverloadedStrings #-}
{-# LANGUAGE TypeApplications #-}
{-# LANGUAGE NoImplicitPrelude #-}

module ADARewardDistribution where

import PlutusTx.Prelude
import PlutusTx
import Plutus.Contract
import Plutus.V1.Ledger.Api
import Plutus.V1.Ledger.Value
import Ledger
import Ledger.Constraints as Constraints
import Playground.Contract
import Prelude (IO, Show(..), String, Integer, ($), (<>), print)

-- | ADA distribution validator
validateADAReward :: Integer -> PubKeyHash -> POSIXTime -> Bool
validateADAReward score user currentTime =
   let expiration = currentTime + 604800  -- 7 days from now
      rewardAmount = case score of
         s | s >= 90 && s <= 100 -> 20000000  -- 20 ADA
         s | s >= 75 && s <= 89  -> 10000000  -- 10 ADA
         s | s >= 60 && s <= 74  -> 5000000   -- 5 ADA
         _ -> 0
   in rewardAmount > 0

-- | Main function to test the validator
main :: IO ()
main = do
   let user = PaymentPubKeyHash "6f1c16d8e5b3a2e5d4d8f3b4d2c6a3f9"
      score = 85
      currentTime = 1694275200  -- Example timestamp (Unix time)
      isValid = validateADAReward score user currentTime
   print $ "Reward validation result: " <> show isValid
```

This approach significantly reduces operational overhead and enhances governance transparency.

(3) Dynamic Optimization of Reward Issuance Logic

In this system, the Metric Analyzer Agent continuously collects and analyzes user score data (e.g., contribution scores, knowledge scores, network scores), which serves as the trigger for the LLM Agent to dynamically optimize the conditions and contents of reward issuance.

The LLM Agent evaluates a combination of factors, including recent user activity history, score trends, and past reward history, to design the most appropriate set of rewards at that moment.

For example, it might propose a combination of rewards like "NFT badge + exclusive tokens" or "discount code + re-entry rights", expressed in natural language, which is then converted into a structured DSL script.

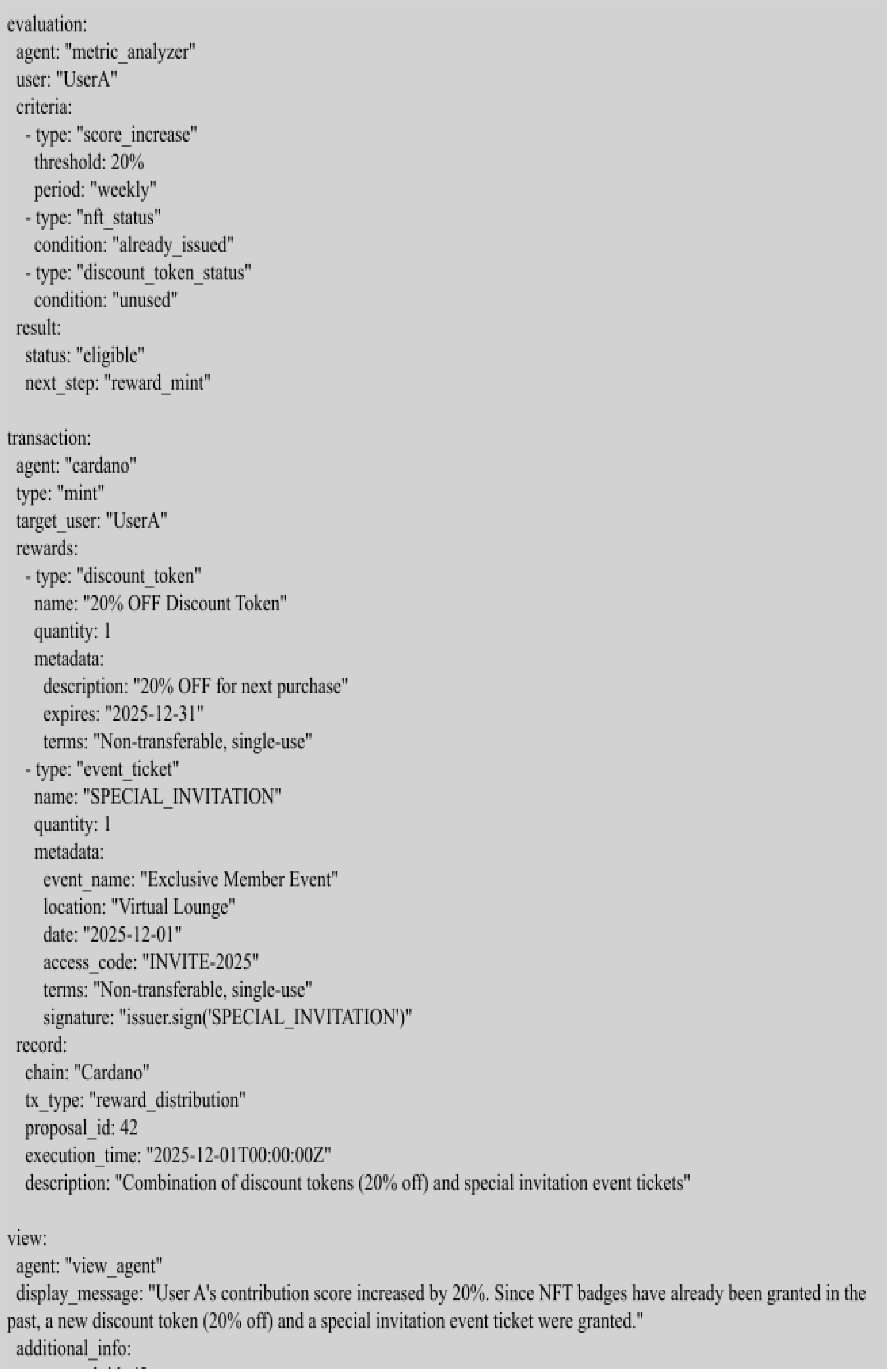
This optimized reward script is then handed over to the Cardano Agent, which automatically executes the actual reward transactions on the Cardano blockchain. This includes token issuance, wallet transfers, and NFT metadata registration.

Example Business Logic:

Consider the following business logic:

"If User A's contribution score has increased by 20% compared to the previous week, and the user already holds an NFT badge but still has unused discount tokens, the Cardano Agent mints a 20% OFF discount token and a special invitation event ticket, and sends them to User A's wallet."

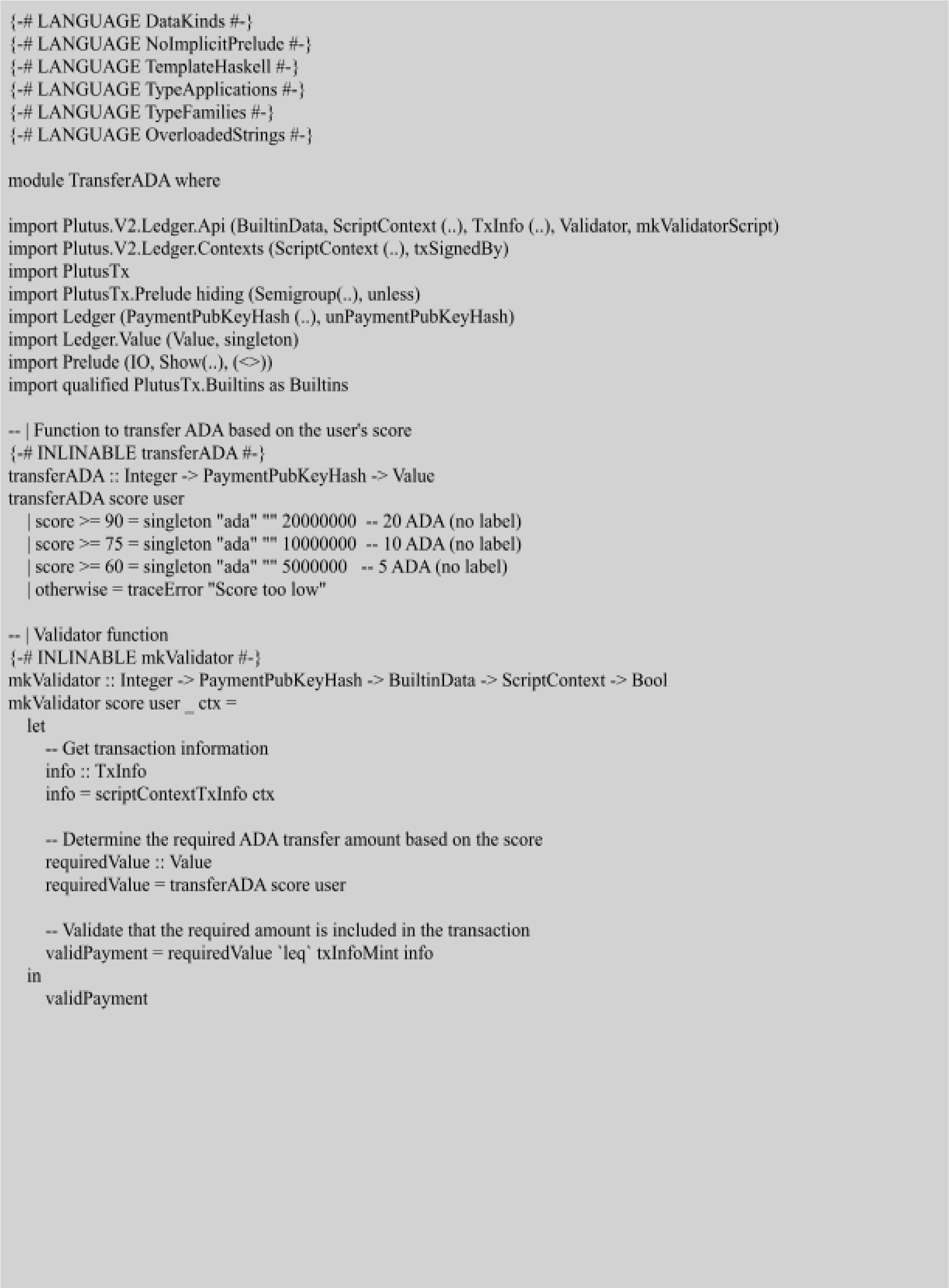
The corresponding DSL for this multi-agent system (instruction to the orchestrator) might look like this:



```
evaluation:
  agent: "metric_analyzer"
  user: "UserA"
  criteria:
    - type: "score_increase"
      threshold: 20%
      period: "weekly"
    - type: "nft_status"
      condition: "already_issued"
    - type: "discount_token_status"
      condition: "unused"
  result:
    status: "eligible"
    next_step: "reward_mint"

transaction:
  agent: "cardano"
  type: "mint"
  target_user: "UserA"
  rewards:
    - type: "discount_token"
      name: "20% OFF Discount Token"
      quantity: 1
      metadata:
        description: "20% OFF for next purchase"
        expires: "2025-12-31"
        terms: "Non-transferable, single-use"
    - type: "event_ticket"
      name: "SPECIAL_INVITATION"
      quantity: 1
      metadata:
        event_name: "Exclusive Member Event"
        location: "Virtual Lounge"
        date: "2025-12-01"
        access_code: "INVITE-2025"
        terms: "Non-transferable, single-use"
        signature: "issuer.sign('SPECIAL_INVITATION')"
  record:
    chain: "Cardano"
    tx_type: "reward_distribution"
    proposal_id: 42
    execution_time: "2025-12-01T00:00:00Z"
    description: "Combination of discount tokens (20% off) and special invitation event tickets"

view:
  agent: "view_agent"
  display_message: "User A's contribution score increased by 20%. Since NFT badges have already been granted in the
past, a new discount token (20% off) and a special invitation event ticket were granted."
  additional_info:
```

Upon receiving this DSL, the Orchestrator Agent issues a command to the previously mentioned Metric Analyzer Agent to verify the user's score.

After confirming the score increase, the Orchestrator Agent then instructs the Cardano Agent with the following commands:



```
{-# LANGUAGE DataKinds #-}
{-# LANGUAGE NoImplicitPrelude #-}
{-# LANGUAGE TemplateHaskell #-}
{-# LANGUAGE TypeApplications #-}
{-# LANGUAGE TypeFamilies #-}
{-# LANGUAGE OverloadedStrings #-}

module TransferADA where

import Plutus.V2.Ledger.Api (BuiltinData, ScriptContext (..), TxInfo (..), Validator, mkValidatorScript)
import Plutus.V2.Ledger.Contexts (ScriptContext (..), txSignedBy)
import PlutusTx
import PlutusTx.Prelude hiding (Semigroup(..), unless)
import Ledger (PaymentPubKeyHash (..), unPaymentPubKeyHash)
import Ledger.Value (Value, singleton)
import Prelude (IO, Show(..), (<>))
import qualified PlutusTx.Builtins as Builtins

-- | Function to transfer ADA based on the user's score
{-# INLINABLE transferADA #-}
transferADA :: Integer -> PaymentPubKeyHash -> Value
transferADA score user
    | score >= 90 = singleton "ada" "" 20000000  -- 20 ADA (no label)
    | score >= 75 = singleton "ada" "" 10000000  -- 10 ADA (no label)
    | score >= 60 = singleton "ada" "" 5000000   -- 5 ADA (no label)
    | otherwise = traceError "Score too low"

-- | Validator function
{-# INLINABLE mkValidator #-}
mkValidator :: Integer -> PaymentPubKeyHash -> BuiltinData -> ScriptContext -> Bool
mkValidator score user _ ctx =
    let
        -- Get transaction information
        info :: TxInfo
        info = scriptContextTxInfo ctx

        -- Determine the required ADA transfer amount based on the score
        requiredValue :: Value
        requiredValue = transferADA score user

        -- Validate that the required amount is included in the transaction
        validPayment = requiredValue `leq` txInfoMint info
    in
        validPayment
```

Cardano Agent Operations

(1) Mint Operations

Token Issuance (token)

- Discount Token:
    - Token Name: DISCOUNT_TOKEN
    - Quantity: 1
    - Recipient: user.wallet
    - Metadata: Description, expiration date, usage conditions
    - Post-Mint: Add the minted token to the user's wallet.
- Ticket Issuance (ticket)
- Special Invitation Event Ticket:
    - Token Name: SPECIAL_INVITATION
    - Quantity: 1
    - Recipient: user.wallet
    - Metadata: Event name, location, date, access code, digital signature
    - Post-Mint: Add the issuer's digital signature to the event ticket and commit it to the blockchain.

Signature Verification

- Verify Issuer Signature:
    - Ensure the issuer's digital signature is correct.
- Metadata Integrity:
    - Validate that the metadata has not been tampered with.

Mint Transaction Creation

- Transaction Preparation:
    - Create the necessary tx (transaction) for minting.
- Fee Calculation:

    - Calculate required fees and minting costs.
- Consistency Check:
    - Verify the consistency of the transaction before broadcasting.

(2) Transaction Logging

Metadata Storage

- Record Details:
    - Log transaction details (txType, chain, description) for traceability.
- Blockchain Archiving:
    - Permanently store the mint contents and history on the Cardano chain.

Archive Update

- Add to History DB:
    - Store token and ticket issuance information in history databases like XTDB.
- Future Audits:
    - Maintain accurate records for future audits and analysis.

Transaction Broadcasting

- Broadcast Transaction:
    - Send the prepared transaction to the Cardano network.
- Status Tracking:
    - Monitor the status until the transaction is confirmed on the blockchain.

User Notification

- Notification Issuance:
    - Notify the user via email or in-app message.
- Special QR Code Generation:
    - Generate special QR codes if the metadata includes access codes or event information.

Transaction Feedback via View Agent

The transaction results are then fed back to the user through the View Agent as follows:

- Discount Token: A 20% OFF token is added to the user's wallet.
- Event Ticket: A Special Invitation Ticket is issued, signed, and stored in the wallet.
- History Management: The issuance history is permanently stored in XTDB for future reference.

Flexible, Personalized Incentive System

This multi-agent transaction process allows for dynamically optimized incentives based on each user's situation, rather than relying on static reward rules.

This flexibility enables personalized incentives that can adapt to user state changes, supporting a strategic, cyclical reward ecosystem designed to sustain and enhance engagement.

(4) On-Chain/Off-Chain Separation and Accountability

This system emphasizes transparency and accountability in the issuance and execution of smart contracts, incorporating a clear separation between on-chain and off-chain functions, supported by a robust information disclosure mechanism.

On-Chain (Plutus) Components:

- The Cardano Agent deploys Plutus-compatible smart contracts for executable reward logic, including:
  - NFT issuance conditions
  - Signature verification rules
  - Token distribution parameters

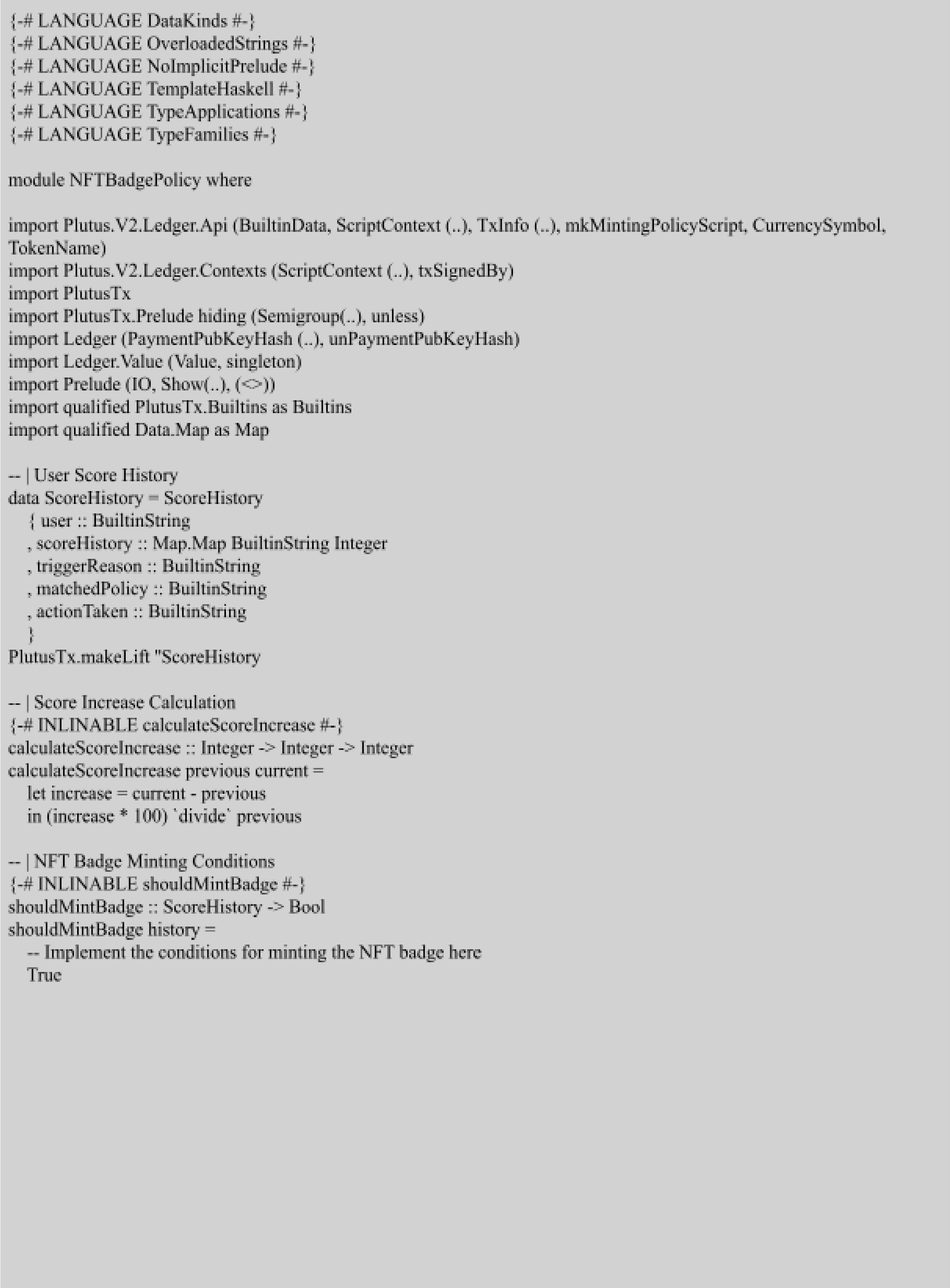
Off-Chain (XTDB, IPFS) Components:

- Complex evaluation histories, score change reasons, reward histories, and detailed user context are managed off-chain.
- These are stored in distributed record systems like XTDB and IPFS, ensuring both flexibility and scalability.

Natural Language Logging by LLM Agent:

- The LLM Agent generates natural language logs explaining why a reward was issued, including:
    - Which user scores and histories met the reward criteria
    - Why the specific reward was triggered
- These logs are then provided to the View Agent for presentation on dashboards, allowing both users and administrators to verify the reasoning behind each reward.
- This ensures transparency, even for automated reward processes, maintaining a high standard of accountability.

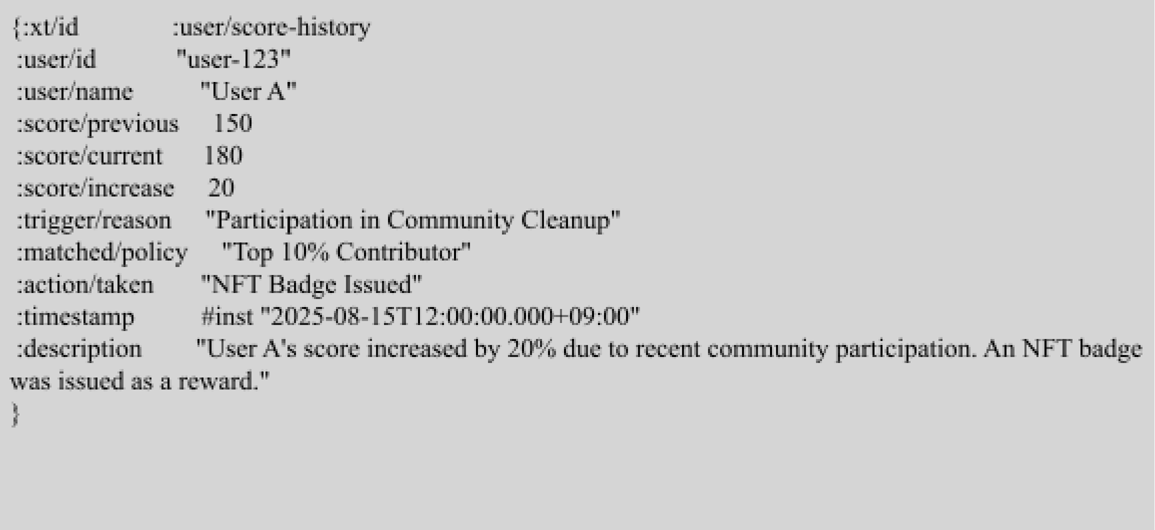
Example:



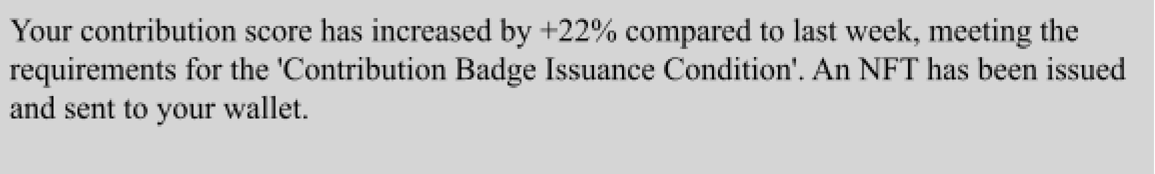
```
{-# LANGUAGE DataKinds #-}
{-# LANGUAGE OverloadedStrings #-}
{-# LANGUAGE NoImplicitPrelude #-}
{-# LANGUAGE TemplateHaskell #-}
{-# LANGUAGE TypeApplications #-}
{-# LANGUAGE TypeFamilies #-}

module NFTBadgePolicy where

import Plutus.V2.Ledger.Api (BuiltinData, ScriptContext (..), TxInfo (..), mkMintingPolicyScript, CurrencySymbol,
TokenName)
import Plutus.V2.Ledger.Contexts (ScriptContext (..), txSignedBy)
import PlutusTx
import PlutusTx.Prelude hiding (Semigroup(..), unless)
import Ledger (PaymentPubKeyHash (..), unPaymentPubKeyHash)
import Ledger.Value (Value, singleton)
import Prelude (IO, Show(..), (<>))
import qualified PlutusTx.Builtins as Builtins
import qualified Data.Map as Map

-- | User Score History
data ScoreHistory = ScoreHistory
    { user :: BuiltinString
    , scoreHistory :: Map.Map BuiltinString Integer
    , triggerReason :: BuiltinString
    , matchedPolicy :: BuiltinString
    , actionTaken :: BuiltinString
    }
PlutusTx.makeLift ''ScoreHistory

-- | Score Increase Calculation
{-# INLINABLE calculateScoreIncrease #-}
calculateScoreIncrease :: Integer -> Integer -> Integer
calculateScoreIncrease previous current =
    let increase = current - previous
    in (increase * 100) `divide` previous

-- | NFT Badge Minting Conditions
{-# INLINABLE shouldMintBadge #-}
shouldMintBadge :: ScoreHistory -> Bool
shouldMintBadge history =
    -- Implement the conditions for minting the NFT badge here
    True
```

Below is an example of the on-chain content recorded in Plutus:

The corresponding off-chain (XTDB) DSL is as follows:



```
{:xt/id            :user/score-history
 :user/id          "user-123"
 :user/name         "User A"
 :score/previous    150
 :score/current     180
 :score/increase    20
 :trigger/reason    "Participation in Community Cleanup"
 :matched/policy    "Top 10% Contributor"
 :action/taken      "NFT Badge Issued"
 :timestamp         #inst "2025-08-15T12:00:00.000+09:00"
 :description       "User A's score increased by 20% due to recent community participation. An NFT badge
was issued as a reward."
}
```



Your contribution score has increased by +22% compared to last week, meeting the requirements for the 'Contribution Badge Issuance Condition'. An NFT has been issued and sent to your wallet.

Example Display by the View Agent:

In this way, by clearly separating the roles of smart contract execution (Cardano) and meaning generation and explanation (LLM and View Agent, off-chain), the system achieves both automation and transparency.

This approach ensures that reward logic is not only automated but also understandable and trustworthy for users, providing a foundation for transparent, data-driven incentive management.

(5) Enhanced Security and Verifiability

This system prioritizes security and verifiability in the automated generation of smart contracts and transaction scripts, adopting a multi-layered verification approach between the LLM Agent and Cardano Agent.

Automated Validation and Constraint Checking

First, the Cardano Agent automatically performs formal validation and constraint checking on the DSL scripts and smart contract code generated by the LLM Agent based on natural language prompts.

This includes:

- Type Consistency

- Syntactic Accuracy
- Logical Consistency

Additionally, the system conducts static analysis to detect smart contract-specific vulnerabilities, such as:

- Double Spending
- Missing Signatures
- Invalid Token Issuance Conditions

High-Risk Operation Controls (HITL - Human-in-the-Loop)

For high-risk operations, such as bulk reward issuance, policy rule changes, or asset transfers to multiple wallets, the system incorporates a HITL (Human-in-the-Loop) process.

This manual approval step is designed for critical actions and includes a dashboard interface that presents:

- Natural language logs
- Reasoning behind the transaction
- Risk levels

Administrators or auditors can approve, reject, or modify these transactions before execution.

Example High-Risk Transaction Considerations

For a transaction task such as distributing a large amount of ADA to all users, multiple risk factors must be considered:

1. Liquidity and Fund Exhaustion
    - Insufficient Balance: Total distribution amount may exceed wallet balance.
    - Liquidity Drain: Large ADA outflow can impact market prices.
    - Fee Shortfall: Insufficient gas fees for transaction processing.
2. Transaction Spam Risk
    - Blockchain Congestion: High transaction volume in a short time can overload the network.

- Mempool Overload: Transaction memory pool overflow can delay other transactions.
- DDoS Attack: Potential for deliberate network disruption through excessive transactions.

3. Double Spending and Replay Attacks
   - Double Spending: Risk of the same ADA being sent multiple times.
   - Replay Attack: Risk of reusing the same transaction multiple times.
   - Resend Exploits: Attackers attempting to reuse past transactions.
4. Configuration Errors and Unintended Behavior
   - Undefined Variables: Errors if user.wallet is not properly defined.
   - Script Errors: Invalid syntax or undefined token definitions.
   - Logic Errors: Unintended loops or branching errors causing multiple payments.
5. Insufficient User Filtering
   - Whitelist Management: Risk of unauthorized addresses receiving funds.
   - Blacklist Ignorance: Potential for fraudulent wallets to receive funds.
   - New Account Abuse: Risk of mass creation of new accounts to exploit rewards.
6. Insider Risk and Privilege Management
   - Misconfigured Permissions: Risk of administrators mistakenly distributing funds to all users.
   - Unauthorized Access: Risk of admin wallet compromise.
   - HITL Bypass: Risk of manual approval process being bypassed.
7. Operational Limits Exceeded
   - Token Supply Limits: Risk of exceeding maximum token supply.
   - Fund Depletion: Risk of exhausting operational funds in a single transaction.
   - Cash Flow Impact: Potential for project-wide financial disruption.
8. Anomalous Scores and Conditions

    - Empty Set Conditions: Risk of impossible conditions like score > 1000.
    - Outlier Values: Risk of extreme scores or invalid user IDs.
    - Condition Bypass: Risk of unauthorized fund transfers through condition manipulation.

9. Front-Running and Economic Attacks

    - Front-Running: Risk of users manipulating transaction order for unfair advantage.
    - Arbitrage: Potential for timing-based profit exploits.
    - Price Manipulation: Risk of significant market impact from large ADA releases.

10. Privacy and Data Leakage

- Address Tracking: Risk of user identity exposure through transaction history.
- Metadata Leakage: Risk of sensitive information being exposed through NFT metadata.
- Transaction Linking: Risk of cross-linking with other transactions, compromising privacy.

Automated Checks by Cardano Agent

For some of these risks, automated checks can be performed by the Cardano Agent, including:

- Liquidity and Balance Checks:
    - Validate wallet balance before transaction execution.
    - Pre-calculate transaction fees to ensure sufficient coverage.
    - Monitor overall system liquidity.
- Spam Prevention:
    - Monitor mempool capacity to prevent overload.
    - Implement rate limits on repeated transactions.
- Double Spend and Replay Prevention:
    - Use unique nonce for each transaction.
    - Reject reuse of the same TxID.

- Configuration and Logic Validation:
    - Ensure user.wallet and tokens are properly defined.
    - Perform pre-execution syntax checks.
- User Filtering:
    - Enforce whitelists for trusted addresses.
    - Maintain blacklists for known fraudulent wallets.

HITL Workflows for Critical Transactions

Certain high-risk transactions require final approval from human operators, such as:

- Liquidity Risk:
    - Manual approval for high-value ADA transfers.
- Configuration Errors:
    - Review of large transactions exceeding normal limits.
- Insider Risk:
    - Manual verification for administrator-level operations.
- Operational Limits:
    - Manual checks for transactions exceeding 95% of token supply limits.
- Economic Attack Prevention:
    - Manual review for high-volume ADA transactions.

Example Workflow for HITL:

1. LLM Agent generates a script: "Distribute 10 ADA to all users"
5. Cardano Agent flags a risk: "Total distribution exceeds 95% of available funds"
6. Transition to HITL: The script is marked as "Pending Admin Approval"
7. Dashboard Display via View Agent:
    - "High-value transaction detected. Target: 2,100 users x 10 ADA = 21,000 ADA. Approve?"
    - Options: ✅ Approve / ✏️ Modify / ❌ Reject

By appropriately integrating automated generation and human review, this approach significantly reduces security risks while enhancing accountability in smart contract operations.

This is particularly critical for public, transparent systems like regional currencies and decentralized reward mechanisms, where trust and auditability are essential for long-term stability.

(6) Advanced Applications and DAO Integration

This system is designed to integrate with Decentralized Autonomous Organizations (DAOs), allowing for the seamless generation and execution of smart contracts based on community-driven decision-making and token economies.

Automated Reward Rule Generation from DAO Proposals

Specifically, the LLM Agent can automatically generate reward rules based on DAO governance votes, converting the natural language proposals submitted by members into structured DSL (Domain-Specific Language) scripts.

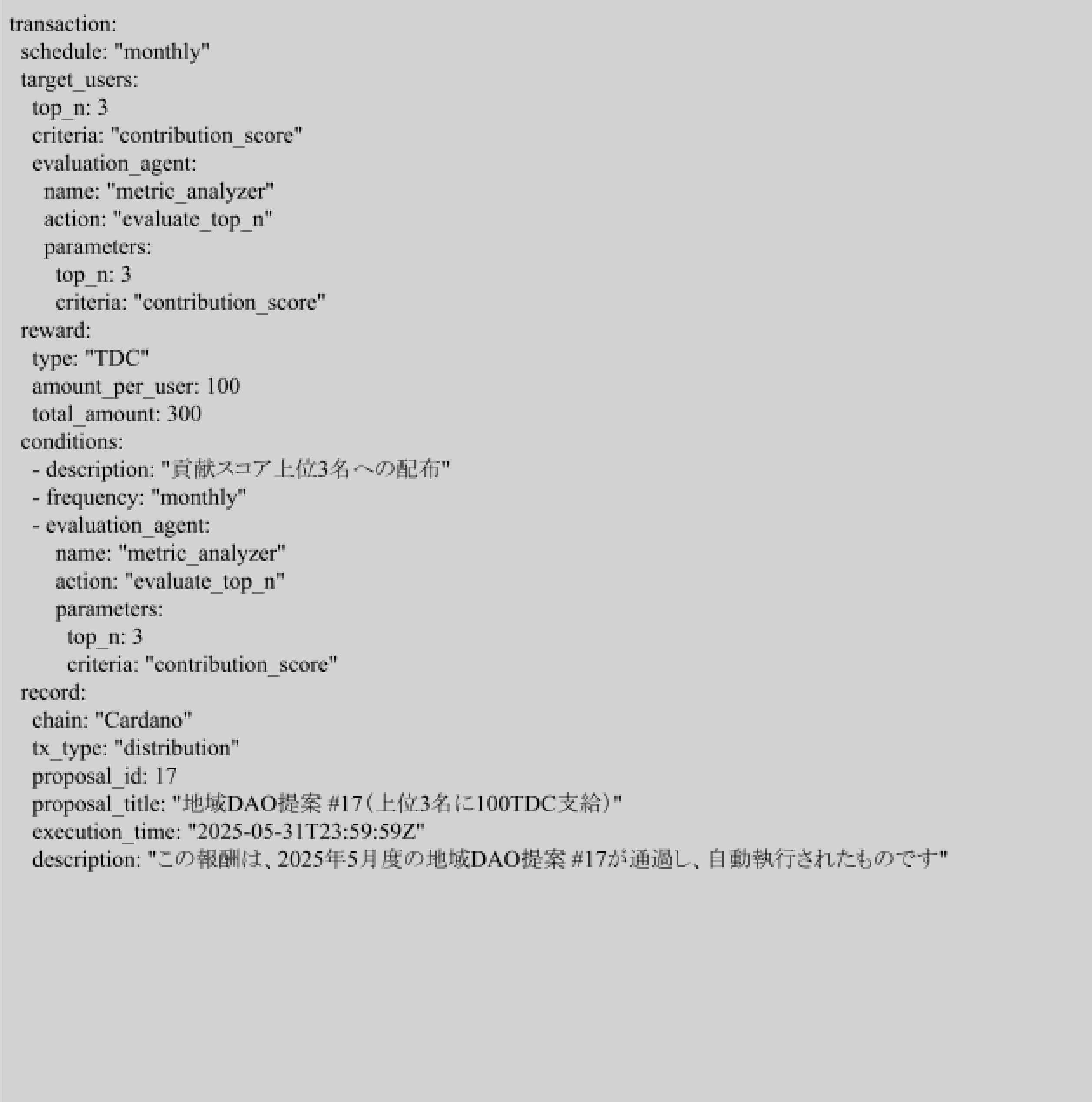
This process involves:

1. Proposal Interpretation: The LLM Agent interprets the conditions, eligibility criteria, and reward mechanismsdescribed in the DAO proposal.
2. DSL Conversion: It then converts these human-defined policies into executable DSL scripts, ensuring the resulting code is both flexible and secure.
3. Blockchain Deployment: The Cardano Agent uses these DSL scripts to execute reward transactions, distribute tokens to wallets, mint NFTs, and record the execution history on-chain, providing immutable proof of the DAO's decisions.

Example Use Case:

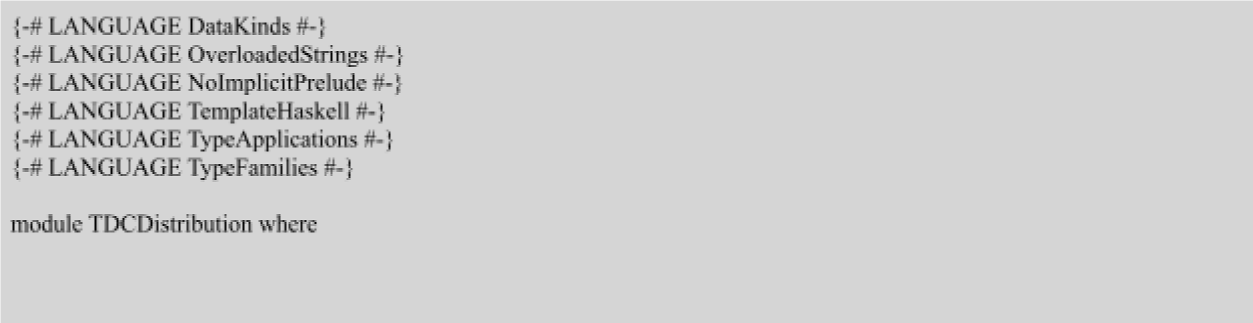
Consider the following DAO-approved proposal:

"Each month, use the Metric Analyzer Agent to identify the top 3 contributors based on their contribution scores. The Cardano Agent then distributes 100 TDC (local tokens) to each of these top 3 contributors."



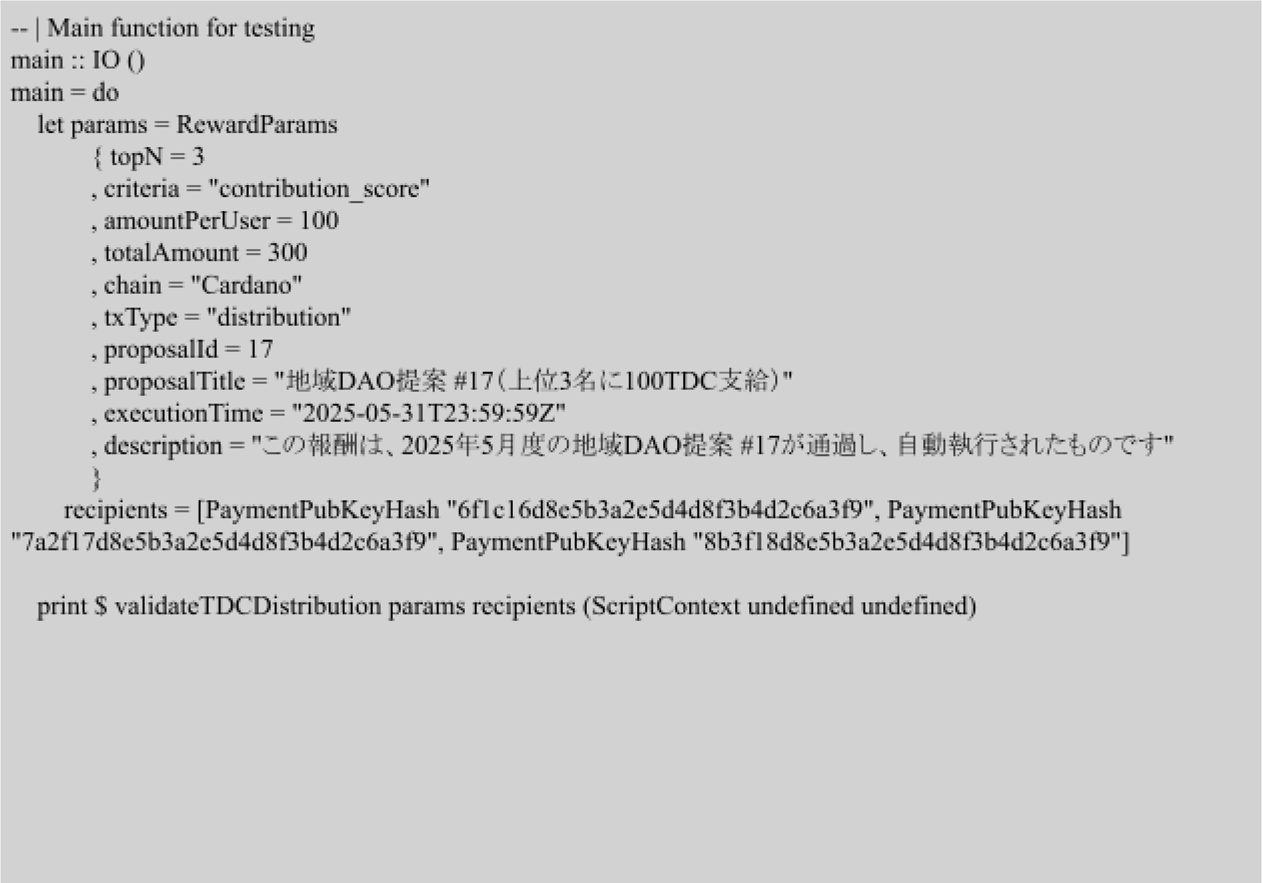
```
transaction:
  schedule: "monthly"
  target_users:
    top_n: 3
    criteria: "contribution_score"
    evaluation_agent:
      name: "metric_analyzer"
      action: "evaluate_top_n"
      parameters:
        top_n: 3
        criteria: "contribution_score"
  reward:
    type: "TDC"
    amount_per_user: 100
    total_amount: 300
  conditions:
    - description: "貢献スコア上位3名への配布"
    - frequency: "monthly"
    - evaluation_agent:
        name: "metric_analyzer"
        action: "evaluate_top_n"
        parameters:
          top_n: 3
          criteria: "contribution_score"
  record:
    chain: "Cardano"
    tx_type: "distribution"
    proposal_id: 17
    proposal_title: "地域DAO提案 #17（上位3名に100TDC支給）"
    execution_time: "2025-05-31T23:59:59Z"
    description: "この報酬は、2025年5月度の地域DAO提案 #17が通過し、自動執行されたものです"
```

The Haskell code that the Cardano Agent would execute based on this DSL is as follows:



```
{-# LANGUAGE DataKinds #-}
{-# LANGUAGE OverloadedStrings #-}
{-# LANGUAGE NoImplicitPrelude #-}
{-# LANGUAGE TemplateHaskell #-}
{-# LANGUAGE TypeApplications #-}
{-# LANGUAGE TypeFamilies #-}

module TDCDistribution where
```

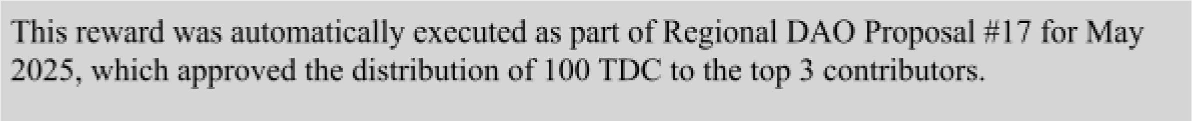
```
import Plutus.V2.Ledger.Api (BuiltinData, ScriptContext (..), TxInfo (..), mkMintingPolicyScript, CurrencySymbol,
TokenName)
import Plutus.V2.Ledger.Contexts (ScriptContext (..), txSignedBy)
import PlutusTx
import PlutusTx.Prelude hiding (Semigroup(..), unless)
import Ledger (PaymentPubKeyHash (..), unPaymentPubKeyHash, CurrencySymbol, TokenName)
import Ledger.Value (Value, singleton)
import Prelude (IO, Show(..), (<>))
import qualified PlutusTx.Builtins as Builtins
import qualified Data.Map as Map

-- | Reward parameters
data RewardParams = RewardParams
  { topN :: Integer
  , criteria :: BuiltinString
  , amountPerUser :: Integer
  , totalAmount :: Integer
  , chain :: BuiltinString
  , txType :: BuiltinString
  , proposalId :: Integer
  , proposalTitle :: BuiltinString
  , executionTime :: BuiltinString
  , description :: BuiltinString
  }
PlutusTx.makeLift "RewardParams

-- | Validate the distribution of TDC tokens to the top N contributors
{-# INLINABLE validateTDCDistribution #-}
validateTDCDistribution :: RewardParams -> [PaymentPubKeyHash] -> ScriptContext -> Bool
validateTDCDistribution params recipients ctx =
  let
    info :: TxInfo
    info = scriptContextTxInfo ctx

    -- Calculate total expected distribution amount
    expectedTotal :: Integer
    expectedTotal = amountPerUser params * topN params

    -- Check if total distribution matches the plan
    totalDistributed :: Integer
    totalDistributed = sum [amountPerUser params | _ <- recipients]

    -- Check if the total amount is correct
    correctAmount = totalDistributed == totalAmount params

    -- Check if the proposal ID is correct
    correctProposal = proposalId params == 17

    -- Ensure the execution time is valid (basic time check)
    validTime = executionTime params == "2025-05-31T23:59:59Z"

  in correctAmount && correctProposal && validTime
```



```
-- | Main function for testing
main :: IO ()
main = do
   let params = RewardParams
         { topN = 3
         , criteria = "contribution_score"
         , amountPerUser = 100
         , totalAmount = 300
         , chain = "Cardano"
         , txType = "distribution"
         , proposalId = 17
         , proposalTitle = "地域DAO提案 #17（上位3名に100TDC支給）"
         , executionTime = "2025-05-31T23:59:59Z"
         , description = "この報酬は、2025年5月度の地域DAO提案 #17が通過し、自動執行されたものです"
         }
      recipients = [PaymentPubKeyHash "6f1c16d8e5b3a2e5d4d8f3b4d2c6a3f9", PaymentPubKeyHash
"7a2f17d8e5b3a2e5d4d8f3b4d2c6a3f9", PaymentPubKeyHash "8b3f18d8e5b3a2e5d4d8f3b4d2c6a3f9"]

   print $ validateTDCDistribution params recipients (ScriptContext undefined undefined)
```

In this case, the directive received within the chain is to "distribute 100 TDC to the top 3 user wallets", but the generated code includes additional "blockchain transaction logging", "data validation", and "testing" as standard operations.

These additional steps ensure secure and verifiable token distribution, automatically integrating the necessary checks for transaction consistency, correct proposal ID, and execution time validation.



This reward was automatically executed as part of Regional DAO Proposal #17 for May 2025, which approved the distribution of 100 TDC to the top 3 contributors.

View Agent Display Example:

In this way, by integrating DAO governance with LLM and Cardano for automated rule implementation, it becomes possible to establish a fair and transparent reward system based on decentralized decision-making.

This approach can be applied to various real-world use cases, including local currencies, participatory policy-making, and user-driven service design, enabling communities to independently manage and reward contributions.

The Cardano Blockchain Integration Agent, when combined with the context-aware and rule-generation capabilities of LLM, serves as a powerful and flexible foundation for explainable reward processing and smart contractdeployment, supporting a wide range of decentralized applications.

3.2.5. Cross-Reality Agent

This system includes a Cross-Reality Agent as a core module for dynamically extracting and integrating user contributions across digital spaces such as Metaverse, Discord, and SNS, as well as real-world activities. This module bridges the gap between virtual and physical interactions, creating a unified view of user contributions across different platforms.

Key Features of the Cross-Reality Agent:

1. Real-World (Physical Space):
   - In-person interactions and experiential events.
   - Utilization of local resources, guided tours at tourist sites, and local specialty tasting events.
   - Real-time incentive distribution (e.g., stamp rallies, quiz rallies).
2. Discord (Online Community):
   - Instant communication and idea sharing.
   - Cross-regional collaboration beyond physical distance.
   - Pre-event information sharing, follow-ups, and feedback accumulation.
   - Enhanced networking and community building.
3. Metaverse (Virtual Space):
   - Immersive virtual experiences and simulations.
   - Virtual tourism and scenario testing.
   - Virtual item and NFT sales, virtual tours, and community events.

Integration of Real and Digital Contributions:

By combining these different channels, the Cross-Reality Agent enables a richer, more diverse engagement environment. It creates a unified user profile by integrating real-world actions with digital interactions, using AI to analyze and score these activities across platforms.

For example, this agent can:

- Track physical event participation, such as local clean-up activities.
- Monitor Discord conversations for idea sharing and community feedback.
- Capture Metaverse interactions, including virtual tours and item purchases.

Applications of the Cross-Reality Agent:

1. Local Tourism and Community Engagement:
    - Participants earn local currency for joining real-world tours or cultural experiences.
    - These experiences are shared on Discord, earning additional points.
    - Pre-visit training and post-event reflections are held in the Metaverse to deepen participant engagement.
2. Community Problem Solving and Cooperative Projects:
    - Disaster preparedness training, infrastructure improvement, cultural heritage preservation, transportation optimization, energy efficiency, and senior care initiatives.
    - Real-world actions are rewarded, while Discord supports problem-solving discussions and feedback loops.
    - Metaverse environments are used for simulation and scenario planning, enhancing collective decision-making.
3. Local Economy and Commercial Activation:
    - In-store purchases, local product sales, and event participation contribute to a community score.
    - Product reviews, store introductions, and coupon sharing on Discord add to the score.
    - Metaverse shops allow for virtual shopping, NFT sales, and try-before-you-buy experiences.
2. Art and Creative Community Development:
    - Physical art events, collaborative projects, and gallery visits are rewarded.
    - Discord supports art contests, collaborative creations, and artwork reviews.

  - Metaverse galleries showcase virtual art collections and NFT purchases.

Example Scenario:

Consider a scenario where participants join a local cultural tour, earn community tokens, share their experiences on Discord for additional points, and participate in a Metaverse event for pre-event learning and post-event reflections.

This approach not only strengthens community ties but also promotes local tourism, cultural awareness, and economic activation.

(1) Scoring, Bot Response Generation, and Analytics via LLM

This system also serves as a platform for analyzing and processing a wide variety of interaction data, including text chats, voice logs, and event participation history from Discord and the Metaverse, using LLM (Large Language Model)technology.

Core Capabilities:

- Data Collection and Preprocessing:
  - Aggregates data from various sources, including user messages, voice recordings, and activity logs.
  - Cleans data, adds metadata, and converts it into a structured format for analysis.
- Context-Aware Attribute Scoring:
  - Quantitatively evaluates user attributes such as engagement (intensity), expertise (knowledge), and network (relationships).
  - Considers temporal patterns and topic shifts for a more precise user profile.
- Real-Time Response and Bot Generation:
  - Generates dynamic responses based on conversation context and user attributes.
  - Incorporates past interactions and overall thread context for more natural dialogue.
- Automated Report Generation and Feedback:
  - Creates summary reports and analytical insights based on scoring results and conversation characteristics.
  - Provides actionable insights for community management and strategy refinement.

Example:

Consider a participant who joins a real-world cultural tour, shares their experiences on Discord for points, and participates in Metaverse pre-event training and post-event reflections.

For instance, the following Discord and Metaverse conversation data could be collected:

- Discord Messages:
    - "I just completed the local tour! Learned so much about the history of this place."
    - "Looking forward to sharing my experiences with the group."
    - "Any tips for making the most of the virtual tour later?"
- Metaverse Interactions:
    - Virtual tour guide role-playing, scenario testing, and collaborative learning.

This data is then structured, scored, and fed into the Metric Analyzer Agent, creating a comprehensive, cross-platform user profile that reflects real and virtual contributions. This profile is used for reward distribution, user ranking, and community engagement optimization.

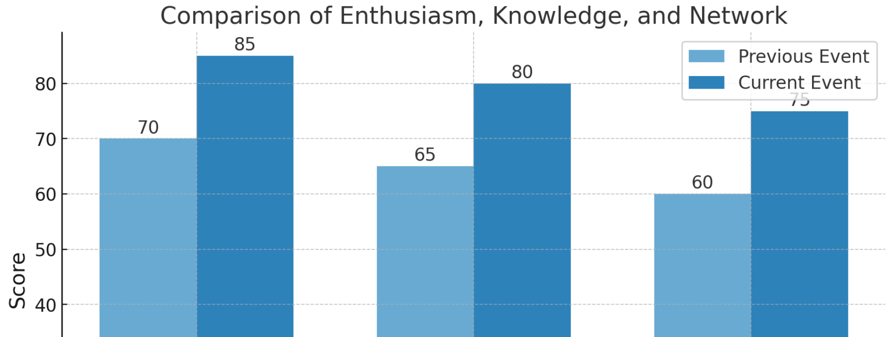
Timestamp	Platform	User	Message
2025-05-07 06:22:02	Metaverse	Eve	Learning about the remains of Sengoku-era castles in the Metaverse was really useful.
2025-05-04 21:05:02	Metaverse	Ivy	It would be fun to have a virtual tour together and then meet up in real life!
2025-05-07 10:08:02	Metaverse	Bob	Seeing the buildings from 3D models in real life is impressive. The handmade stone lanterns were particularly amazing.
2025-05-06 10:08:02	Discord	Hannah	Today's history tour was really enjoyable!
2025-05-05 14:07:02	Discord	Alice	I learned so much about local culture! The ancient rituals and their roles were especially fascinating.
2025-05-02 22:30:02	Metaverse	Bob	Let's have another reflection event in the Metaverse! It was great to connect with new people.
2025-05-07 01:19:02	Metaverse	Dave	It would be fun to have a virtual tour together and then meet up in real life!
2025-05-01 21:32:02	Discord	Alice	It was great talking with everyone today!
2025-05-07 18:52:02	Discord	Ivy	Today's history tour was really enjoyable!
2025-05-01 20:10:02	Discord	Frank	I uploaded some photos! The view from the mountain was amazing.

2025-05-07 23:12:02	Discord	Alice	I uploaded some photos! These are from the local market. The lively morning atmosphere is wonderful.
2025-05-04 14:24:02	Metaverse	Hannah	Let's have another reflection event in the Metaverse! It will boost interest for the next tour.
2025-05-05 17:13:02	Discord	Charlie	I definitely want to join next time, too.

Based on this data, the LLM Agent analyzed the user intent, sentiment, enthusiasm score, knowledge score, and network score, resulting in the following analysis:

Message	Intent	Sentiment	Enthusiasm	Knowledge	Network
Eve: Learning about Sengoku-era castle ruins in the Metaverse was really useful.	Metaverse Experience	Neutral	3	3	3
Ivy: It would be fun to have a virtual tour together and then meet up in real life!	Tour Guidance	Neutral	3	3	3
Bob: Seeing the buildings from 3D models in real life is impressive. The handmade stone lanterns were particularly amazing.	Others	Positive	3	3	3
Hannah: Today's history tour was really enjoyable!	Tour Guidance	Positive	3	3	3
Alice: I learned a lot about local culture! The ancient rituals and their roles were particularly impressive.	Culture & History Sharing	Neutral	3	3	3
Bob: Let's have another reflection event in the Metaverse! It was great to connect with new people.	Metaverse Experience	Positive	3	3	3
Dave: It would be fun to have a virtual tour together and then meet up in real life!	Tour Guidance	Neutral	3	3	3
Alice: It was great talking with everyone today!	Others	Neutral	3	3	3
Ivy: Today's history tour was really enjoyable!	Tour Guidance	Positive	3	3	3
Frank: I uploaded some photos! This is a stunning view from the mountain.	Photo Sharing	Neutral	3	3	3
Alice: I uploaded some photos! These are from the local market. The bustling morning atmosphere really captures the local charm.	Photo Sharing	Neutral	3	3	3

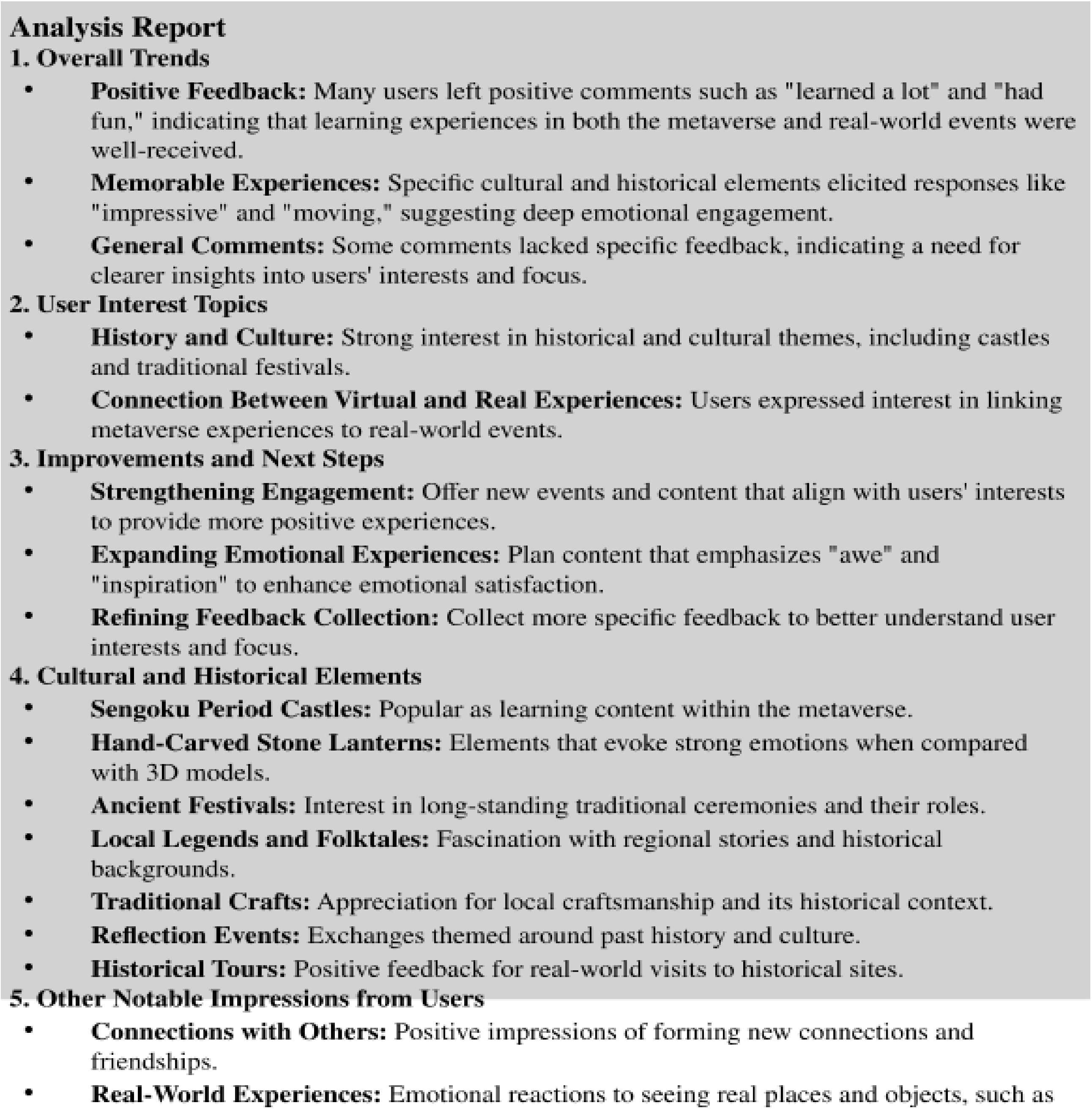
Using these scores, the Metric Analyzer Agent can process the data to implement targeted actions (e.g., reward distribution) for each user. Additionally, it is possible to compare these



scores with previous event data to evaluate trends and engagement improvements, as shown in the example below.

Additionally, using the extracted user intents, sentiments, and Message data, the LLM Agent can generate bot responses that take into account past interactions and the overall context of the conversation thread, as shown in the examples below.

Message	Bot Response
Eve: Learning about Sengoku-era castle ruins in the Metaverse was really useful.	That sounds like a great experience! What did you enjoy the most? Looking forward to seeing you in the next event!
Ivy: It would be fun to have a virtual tour together and then meet up in real life!	Tours are always exciting! Do you have any ideas for the next destination?
Bob: Seeing the buildings from 3D models in real life is impressive. The handmade stone lanterns were particularly amazing.	Thanks for sharing such a wonderful experience! We'd love to hear more about it!
Hannah: Today's history tour was really enjoyable!	Tours are always exciting! Do you have any ideas for the next destination?



Analysis Report

1. Overall Trends

- **Positive Feedback:** Many users left positive comments such as "learned a lot" and "had fun," indicating that learning experiences in both the metaverse and real-world events were well-received.
- **Memorable Experiences:** Specific cultural and historical elements elicited responses like "impressive" and "moving," suggesting deep emotional engagement.
- **General Comments:** Some comments lacked specific feedback, indicating a need for clearer insights into users' interests and focus.

2. User Interest Topics

- **History and Culture:** Strong interest in historical and cultural themes, including castles and traditional festivals.
- **Connection Between Virtual and Real Experiences:** Users expressed interest in linking metaverse experiences to real-world events.

3. Improvements and Next Steps

- **Strengthening Engagement:** Offer new events and content that align with users' interests to provide more positive experiences.
- **Expanding Emotional Experiences:** Plan content that emphasizes "awe" and "inspiration" to enhance emotional satisfaction.
- **Refining Feedback Collection:** Collect more specific feedback to better understand user interests and focus.

4. Cultural and Historical Elements

- **Sengoku Period Castles:** Popular as learning content within the metaverse.
- **Hand-Carved Stone Lanterns:** Elements that evoke strong emotions when compared with 3D models.
- **Ancient Festivals:** Interest in long-standing traditional ceremonies and their roles.
- **Local Legends and Folktales:** Fascination with regional stories and historical backgrounds.
- **Traditional Crafts:** Appreciation for local craftsmanship and its historical context.
- **Reflection Events:** Exchanges themed around past history and culture.
- **Historical Tours:** Positive feedback for real-world visits to historical sites.

5. Other Notable Impressions from Users

- **Connections with Others:** Positive impressions of forming new connections and friendships.
- **Real-World Experiences:** Emotional reactions to seeing real places and objects, such as

Furthermore, by combining these data points, it is possible to automatically generate analytical reports like the one shown below.

Here, overall message trends, user interest topics, and improvement comments for future events are summarized.
In this way, context understanding and classification by LLMs go beyond simple categorization tasks, functioning as an advanced information processing framework that can visualize and structure semantic contributions from each user, supporting subsequent scoring and strategic alignment.

(2) Network Analysis Using GNN

This system leverages Graph Neural Networks (GNNs) to perform precise analysis of dynamic network structures and influence, capturing characteristics that large language models (LLMs) alone may overlook. Specifically, it calculates the following key structural metrics based on connection data derived from user interactions and conversation histories:

1. Degree Centrality (Number of Connections)
   Measures the number of direct connections each user has, identifying highly active participants and influential users within the network.

2. Closeness Centrality (Proximity)
   Evaluates the average shortest path length from a user to all other users, indicating the speed of information spread and the breadth of influence within the network.

3. Community Detection (Conversation Cluster Extraction)
   Identifies naturally emerging topic groups based on reply chains and co-occurrence relationships, allowing visualization and classification of community structures.

4. Betweenness Centrality (Bridge Role)
   Assesses the frequency with which a user acts as an intermediary between different clusters, highlighting users who facilitate cross-community communication.

5. Influence Score (Impact Score)
   Quantifies the extent to which a user's statements and actions influence others, providing a measure of their overall impact on the network.

6. Activation Propagation (Behavior Spread)
   Models how specific actions (e.g., posts, participation, replies) propagate through the network, analyzing patterns of cascading influence.

These metrics are not just indicators of user presence or connection volume, but essential components for understanding the centrality structure and information diffusion patterns within the entire network. The resulting analysis supports a wide range of practical applications, including:

- Contribution Assessment: Identifying highly active or influential users.

- Targeted Campaign Design: Planning effective engagement strategies and outreach campaigns.

- Bot Guidance Optimization: Designing automated responses or guides centered around influential nodes.

Use Case: Local Industry Promotion and Regional Economic Revitalization

To illustrate this approach, consider a use case focused on expanding sales channels for local industries and promoting tourism, aimed at strengthening regional economic activity and cross-industry collaboration. The dataset spans a four-month period from January 2025 to April 2025, involving 500 local residents and tourism businesses. Key components include:

- Users: A total of 500 participants, including local residents, farmers, manufacturers, and tourism operators, actively engaged in sharing proposals and exchanging ideas for local industry growth.

- Nodes: 500 unique nodes representing individual users, with each node capturing a user's statements and interactions.

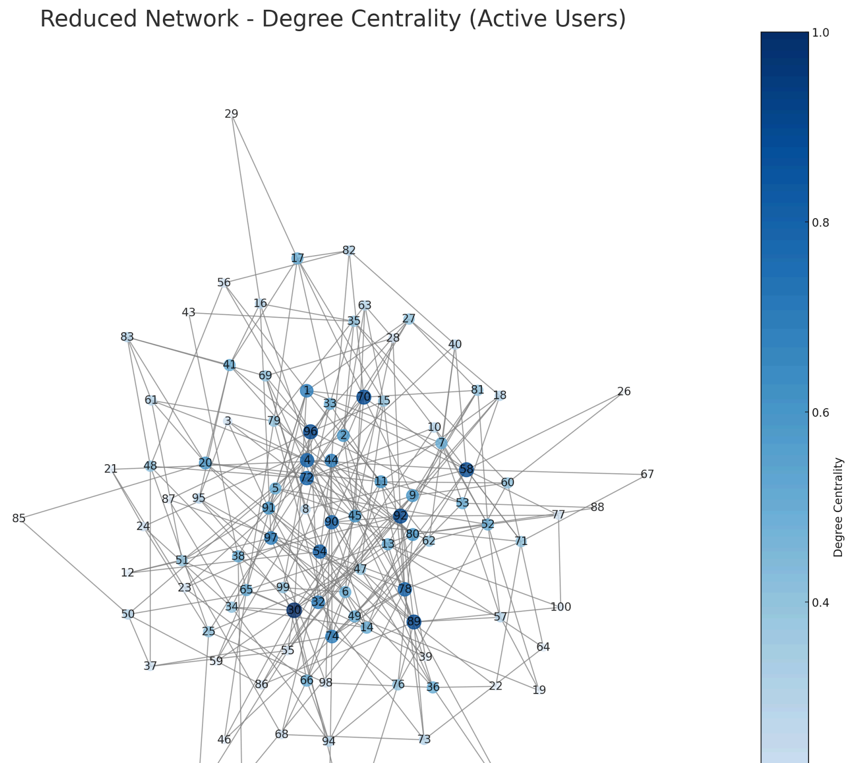
- Edges: 2,000 edges representing direct interactions, feedback, and proposals between users, reflecting inter-industry collaboration.

- Attributes: Each user is assigned attributes such as region, industry type (e.g., agriculture, manufacturing, tourism), posting frequency, and influence score, supporting role identification and influence evaluation within the network.

- Time Period: Data collected from January 2025 to April 2025, incorporating seasonal variations and event-driven impacts.

Example: Degree Centrality Evaluation

The degree centrality evaluation for this dataset, which measures the number of direct connections each user has, provides insights into the most active participants and influential nodes within the regional economic network.



This network diagram visualizes the Degree Centrality of a regional industry revitalization network, comprising local residents and tourism businesses. Each node represents an individual user, while the edges connecting these nodes indicate direct interactions, proposals, and feedback exchanges. The key features of this diagram are as follows:

Key Features of the Degree Centrality Network Diagram

- Node Size:
  The size of each node reflects the frequency of interactions a user has with other members. Larger nodes indicate users who actively contribute through frequent posts and proposals.

- Node Color:
  The color of each node is determined by its Degree Centrality, with more active users represented by darker shades of blue, visually emphasizing their central role within the network.

- Network Structure:
  The overall structure reveals loosely connected clusters, highlighting the critical role certain users play in bridging multiple communities, serving as connectors and facilitators within the network.

Applications of Degree Centrality in Regional Industry Revitalization

1. Identifying Key Members:
   Identify users who frequently contribute and propose ideas (large nodes) to plan leadership and influence-based initiatives.
   Example: Selecting core members for regional product promotion or tourism event planning.

2. Understanding Network Structure:
   Gain insights into the overall network structure to improve information dissemination and strengthen collaborative relationships.
   Example: Identifying central members to facilitate cross-industry communication between agriculture, manufacturing, and tourism sectors.

3. Designing Targeted Strategies:
   Create marketing and community engagement strategies focused on central users.
   Example: Initiating social media campaigns centered on influential users to amplify outreach.

4. Strengthening Connections and Promoting Information Flow:
   Maximize the ripple effect of information spread around active users, encouraging new member participation and cross-industry collaboration.
   Example: Enhancing promotional strategies for new products or expanding tourism routes.

Top 10 High-Degree Centrality Users (Most Active Participants)

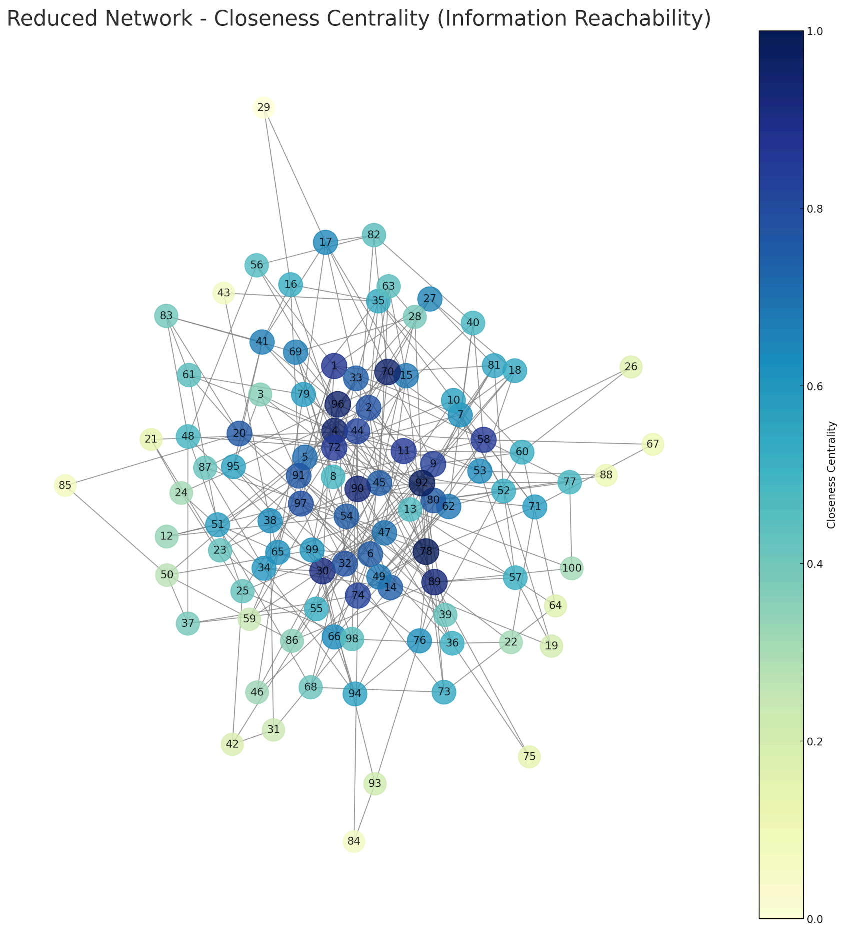
The following are the top 10 most active users with the highest Degree Centrality scores, indicating significant influence within the network:

- User 30 - Degree Centrality: 0.121
- User 58 - Degree Centrality: 0.111
- User 70 - Degree Centrality: 0.111
- User 89 - Degree Centrality: 0.111
- User 92 - Degree Centrality: 0.111
- User 96 - Degree Centrality: 0.111
- User 4 - Degree Centrality: 0.101
- User 54 - Degree Centrality: 0.101
- User 72 - Degree Centrality: 0.101
- User 78 - Degree Centrality: 0.101

These users have a high number of direct connections, making them central figures in the network. In particular, User 30stands out with the highest Degree Centrality, indicating a highly active participant who frequently shares ideas and proposals.

By leveraging Degree Centrality, it is possible to visualize not only the frequency of user interactions but also the overall influence and relationships within the network, enabling more effective strategy planning for regional revitalization.

Next, the results for Closeness Centrality are presented below.



This diagram illustrates the Closeness Centrality analysis results for the network, highlighting the following key features:

Key Features of the Closeness Centrality Network Diagram

- Node Size:
  Nodes are sized based on their Closeness Centrality scores, with larger nodes representing users who have a shorter average distance to all other users. This indicates that these users can efficiently spread information across the network.

- Node Color:
  Nodes with higher Closeness Centrality are displayed in darker teal, reflecting their higher influence and faster information reach within the overall network.

- Network Structure:
  Nodes located near the center of the network play a critical role in facilitating efficient information flow, acting as key conduits for communication across the network.

Top Users with High Closeness Centrality

The users with the highest Closeness Centrality, indicating a shorter average distance to other members and faster information propagation, are as follows:

- User 78 - Closeness Centrality: 0.430
- User 92 - Closeness Centrality: 0.429
- User 4 - Closeness Centrality: 0.425
- User 70 - Closeness Centrality: 0.425
- User 96 - Closeness Centrality: 0.425
- User 89 - Closeness Centrality: 0.421
- User 90 - Closeness Centrality: 0.421
- User 30 - Closeness Centrality: 0.419
- User 1 - Closeness Centrality: 0.414
- User 11 - Closeness Centrality: 0.411

These users are positioned closer to other members in the network, enabling faster information dissemination and acting as central hubs for communication. In particular, User 78 stands out as having the highest Closeness Centrality, indicating a critical role in maintaining efficient information flow throughout the network.

Comparison with Degree Centrality

Previously, User 30 had the highest Degree Centrality, while User 78 now has the highest Closeness Centrality. This difference highlights an important distinction in network dynamics:

- High Degree Centrality (e.g., User 30)
  Indicates users with a large number of direct connections, making them effective focal points for direct outreach and collaboration.
  Example Use Case: In a sales network, these users can serve as key distribution points for direct product promotions.

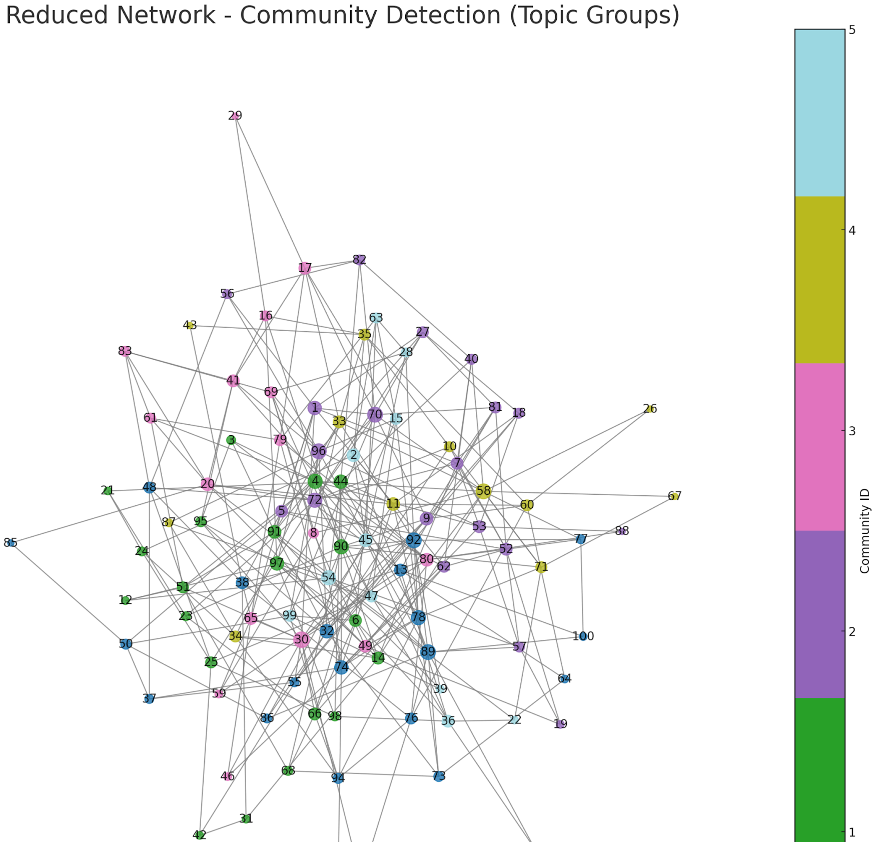
- High Closeness Centrality (e.g., User 78)
  Indicates users who can rapidly distribute information across the entire network due to their short average path length to other members.
  Example Use Case: These users are ideal for rapid dissemination of new product information or promotional campaigns.

Use Cases in Community Analysis

- High Degree Centrality:
  Effective for leadership roles and consensus building, as these users have extensive direct connections.
  Example: Community leaders who can coordinate activities or organize events.

- High Closeness Centrality:
  Suitable for rapid decision-making and efficient information sharing, given their central position in the network.
  Example: Members who can quickly broadcast announcements or mobilize groups for coordinated action.

Next, the analysis results based on Community Detection will be presented, focusing on the identification of naturally emerging topic groups based on user replies and co-occurrence relationships, providing a detailed view of the community structure within the network.



Reduced Network - Community Detection (Topic Groups)

This network contains six major communities, with each node sized based on Degree Centrality, emphasizing users who actively contribute through frequent posts and proposals.

Betweenness Centrality Analysis (Bridging Roles)

Applying Betweenness Centrality analysis to this data reveals key members who act as critical bridges between different clusters, facilitating cross-community communication. The top users identified as essential connectors are as follows:

- User 30 - Betweenness Centrality: 0.060
- User 89 - Betweenness Centrality: 0.059
- User 58 - Betweenness Centrality: 0.056
- User 92 - Betweenness Centrality: 0.055
- User 96 - Betweenness Centrality: 0.049
- User 90 - Betweenness Centrality: 0.047
- User 74 - Betweenness Centrality: 0.044
- User 4 - Betweenness Centrality: 0.043
- User 70 - Betweenness Centrality: 0.042
- User 54 - Betweenness Centrality: 0.042

From this analysis, it is clear that User 30, who previously ranked highest in Degree Centrality, also plays a central role in connecting different communities, acting as a critical bridge within the network.

Interestingly, User 89 also ranks high in Betweenness Centrality despite having a relatively lower Degree Centralityscore. This suggests that User 89 plays a strategic role in connecting otherwise distant clusters, serving as an important intermediary despite having fewer direct connections.

Influence Score Analysis (Network-Wide Impact)

Next, the Influence Score analysis identifies the users with the greatest overall impact on the network, reflecting their ability to significantly shape decisions and influence other members:

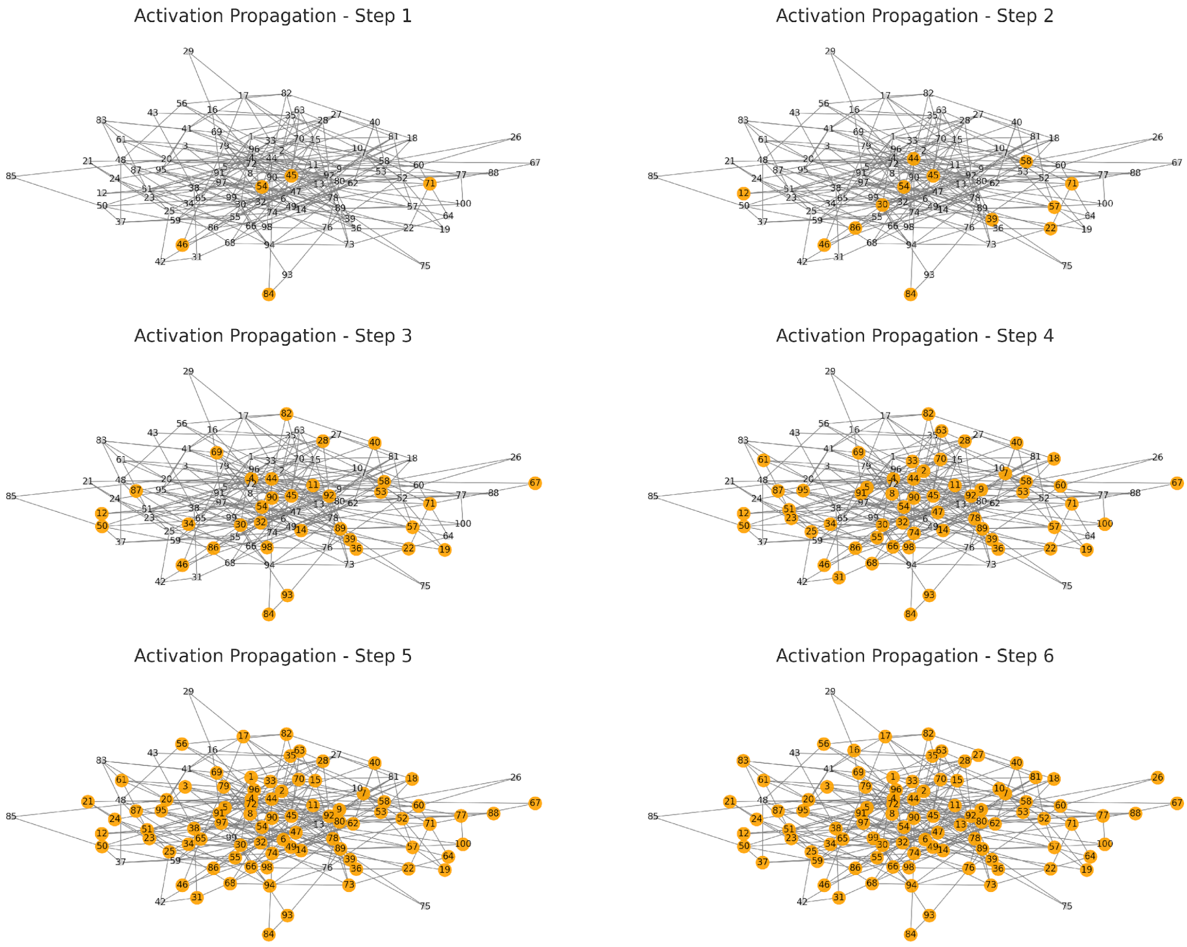
- User 30 - Influence Score: 0.019
- User 58 - Influence Score: 0.018
- User 89 - Influence Score: 0.018
- User 92 - Influence Score: 0.017
- User 96 - Influence Score: 0.017
- User 70 - Influence Score: 0.017
- User 54 - Influence Score: 0.016
- User 90 - Influence Score: 0.016
- User 72 - Influence Score: 0.016
- User 78 - Influence Score: 0.016

These users exert strong influence across the entire network, playing critical roles in information diffusion and decision-making. Notably, User 30 stands out with the highest Influence Score, confirming their central importance within the overall network structure.

Activation Propagation Analysis (Behavior Spread)

Finally, the Activation Propagation analysis simulates how specific actions (e.g., posts, participation, replies) cascade through the network. The diagram illustrates the spread of active nodes over several steps, showing how information initiated by the first five users propagates to other members.

- Node Color: Orange indicates active nodes, while gray represents inactive nodes.



Activation Propagation - Step 1

Activation Propagation - Step 2

Activation Propagation - Step 3

Activation Propagation - Step 4

Activation Propagation - Step 5

Activation Propagation - Step 6

- Propagation Dynamics: As the steps progress, more nodes become active, reflecting the spreading influence across the network.

This visualization provides a clear view of how information can ripple through the network, highlighting the users who play pivotal roles in initiating and sustaining communication waves.

In this way, network structure analysis using GNNs enables both the visualization of dynamic community dynamics and contribution patterns, supporting a shift from mere observation to strategic management.

(3) Hybrid Score Generation Combining LLM and GNN Analysis

To evaluate user contributions from multiple perspectives with high precision, this system adopts a hybrid score generation method that integrates semantic understanding from LLMs (Large Language Models) and structural influence evaluation from GNNs (Graph Neural Networks). This approach enables a dynamic and comprehensive assessment of contributions that goes beyond the limitations of traditional single-metric evaluations.

Key Scores and Their Analysis

- Enthusiasm and Knowledge Spread:
  Evaluates the extent to which statements identified by the LLM as related to

enthusiasm or knowledge propagate across different clusters, leveraging the structural analysis capabilities of the GNN.

- Network Contribution:
  Assesses the frequency with which a user acts as a central node in the conversation network (e.g., high Degree Centrality or Betweenness Centrality), along with the scope of their influence.

- Empathy Formation:
  Integrates the density of emotional reactions such as "Thank you" or "Support" received in response to a user's statements, along with the diversity of nodes expressing these reactions.

Data Recording and Advanced Applications

These scores are recorded and tracked through the Metric Analyzer Agent in the bitemporal database XTDB, accumulating as time-series contribution histories for each user. This enables advanced applications such as:

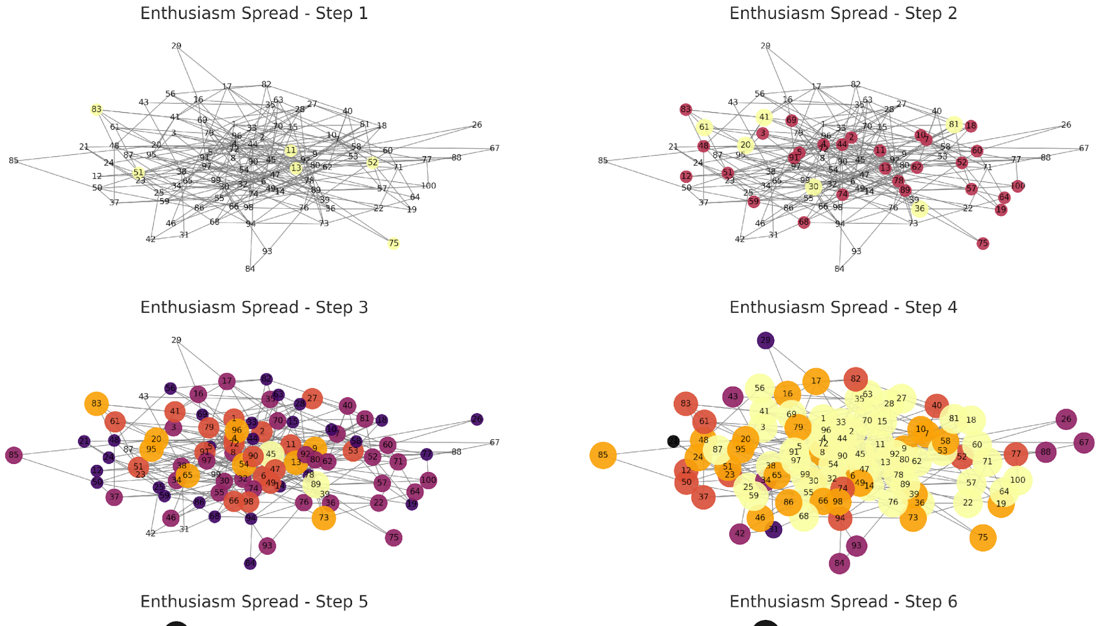
- Optimization of Token/NFT Reward Systems:
  Efficiently distributing rewards based on user scores.

- Visualization of Contribution Behavior:
  Providing real-time insights into user engagement and influence through dashboards.

- Analysis of Contribution Profile Evolution:
  Tracking changes in individual user profiles over time to understand long-term engagement trends.

In this way, the hybrid scoring approach combining LLM and GNN methods forms a powerful foundation for sustained community activation and contribution enhancement, capturing both qualitative and structural aspects of user impact.

Next, an example of Enthusiasm Score Spread Analysis is presented below.



Enthusiasm Spread - Step 1

Enthusiasm Spread - Step 2

Enthusiasm Spread - Step 3

Enthusiasm Spread - Step 4

Enthusiasm Spread - Step 5

Enthusiasm Spread - Step 6

Enthusiasm Spread Score Simulation

This diagram represents the simulation results for the Enthusiasm Spread Score. Six nodes, each selected as initial enthusiasm nodes from different clusters, serve as the starting points for enthusiasm propagation. The diagram visualizes how enthusiasm spreads across clusters over multiple steps.

- Propagation Dynamics:
  At each step, enthusiasm spreads from the initial enthusiasm nodes to adjacent nodes, gradually extending across multiple clusters as time progresses. This results in an increasing number of nodes spanning different clusters, reflecting the widespread distribution of enthusiasm throughout the network.

- Node Color and Size:
  The color intensity of each node indicates the amount of enthusiasm absorbed from different clusters, with darker and larger nodes representing users that have absorbed enthusiasm from multiple clusters.
  Nodes positioned closer to the center of the network tend to absorb enthusiasm from a wider variety of clusters, indicating their critical role in overall enthusiasm diffusion.

Top Users with High Enthusiasm Spread

The users with the highest enthusiasm spread, reaching across six clusters, are as follows:

- User 9 - Enthusiasm Spread Range: 6 clusters
- User 71 - Enthusiasm Spread Range: 6 clusters
- User 83 - Enthusiasm Spread Range: 6 clusters
- User 29 - Enthusiasm Spread Range: 6 clusters
- User 64 - Enthusiasm Spread Range: 6 clusters
- User 1 - Enthusiasm Spread Range: 6 clusters
- User 6 - Enthusiasm Spread Range: 6 clusters
- User 51 - Enthusiasm Spread Range: 6 clusters
- User 82 - Enthusiasm Spread Range: 6 clusters
- User 5 - Enthusiasm Spread Range: 6 clusters

These users exhibit strong enthusiasm propagation effects, significantly influencing topics across a wide range of clusters, and serve as central figures in the network.

Network Contribution Score Analysis

Next, the Network Contribution Score is presented. This score is calculated by combining Degree Centrality and Betweenness Centrality, reflecting both the frequency with which a node acts as a central participant and the overall scope of its influence within the network.

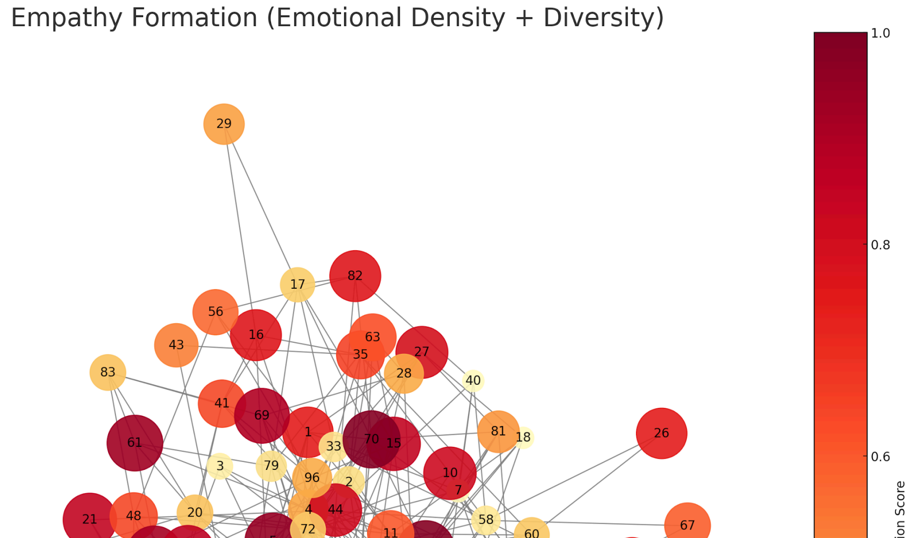
The top 10 users with the highest Network Contribution Scores are as follows:

- User 30 - Contribution Score: 0.097
- User 89 - Contribution Score: 0.090
- User 58 - Contribution Score: 0.089
- User 92 - Contribution Score: 0.089
- User 96 - Contribution Score: 0.086
- User 70 - Contribution Score: 0.084
- User 90 - Contribution Score: 0.079
- User 4 - Contribution Score: 0.078
- User 54 - Contribution Score: 0.077
- User 78 - Contribution Score: 0.077

In particular, User 30 stands out with the highest Network Contribution Score, indicating a strong central role in the network. This user not only maintains numerous direct connections but also plays a key role in facilitating information propagation across different clusters.

Empathy Formation Score Analysis

Finally, the analysis based on Empathy Formation Score is presented below. This score captures the extent to which a user elicits supportive and emotionally positive responses, such as "Thank you" or "Support," from a diverse range of other users within the network.



Empathy Formation Score Analysis

This score evaluates the density of emotional reactions (e.g., "Thank you," "Support") received by each node, along with the diversity of the nodes it is connected to. It provides a comprehensive measure of a user's emotional resonance within the network, integrating both the volume and diversity of positive feedback from other users.

- Node Size:
  Nodes are sized based on their Empathy Formation Scores, with larger nodes representing users who receive a high volume of diverse emotional reactions.
- Node Color:
  Nodes with higher scores are colored more intensely, highlighting users who have received significant support and built diverse connections across different clusters. These users are typically positioned at the center of the network, reflecting their strong emotional influence.

Top 10 Users with High Empathy Formation Scores

The top 10 users with the highest Empathy Formation Scores are as follows:

- User 45 - Empathy Formation Score: 13.0
- User 9 - Empathy Formation Score: 12.8
- User 70 - Empathy Formation Score: 12.8
- User 5 - Empathy Formation Score: 12.4
- User 62 - Empathy Formation Score: 12.4
- User 61 - Empathy Formation Score: 12.2

- User 64 - Empathy Formation Score: 12.2
- User 65 - Empathy Formation Score: 12.0
- User 87 - Empathy Formation Score: 12.0
- User 69 - Empathy Formation Score: 11.8

Notably, User 45, who did not appear prominently in previous analyses, stands out as the user with the highest Empathy Formation Score. This suggests that User 45 is a central figure in fostering emotional connections, receiving widespread support from a diverse range of users across multiple clusters. This high score reflects not only strong direct connections but also a deep emotional impact on the broader community, contributing significantly to overall community engagement.

Integrated Contribution Scoring for Comprehensive Community Analysis

By combining meaning-based evaluations (e.g., enthusiasm and knowledge spread) with structural analyses (e.g., network contribution and empathy formation), this approach provides a more holistic assessment of user contributions. It captures complex dynamics such as "enthusiasm and knowledge diffusion," "network influence," and "emotional resonance," creating a unified framework where evaluation and rewards are consistently aligned.

(4) Automating Contribution Guidance with LLMs

To promote proactive community engagement, this system leverages Large Language Models (LLMs) to analyze user behavior histories, score trends, and structural roles within the network (e.g., central users, bridge nodes, supporters) in real-time. By generating context-aware prompts for "next high-impact actions," LLMs enable users to make autonomous, meaningful contributions without requiring intensive observation or intervention from community managers.

Example Context-Sensitive Contribution Suggestions

LLMs can generate natural, context-sensitive prompts based on each user's activity and network position, such as:

- Welcoming New Members:
  "A new user has joined the cluster where you frequently contribute. Why not send a welcome message?"
- Deepening Discussions:
  "Your recent posts have been receiving a lot of engagement. How about starting a related thread to dive deeper into the topic?"

- Offering Continued Support:
  "A user you helped in the past seems to be facing a challenge again. Would you like to check in and offer support?"

These suggestions are designed to be seamlessly integrated into a user’s natural communication flow, encouraging self-directed contributions while promoting overall community vitality and decentralized support dynamics.

Adaptive and Personalized Prompt Generation

The content, frequency, and timing of these prompts are dynamically adjusted based on each user’s motivations and participation patterns, reducing the risk of intrusive guidance while maximizing the likelihood of meaningful interactions. This approach transforms LLMs from mere conversation generators into agent-like entities capable of both structural understanding and action guidance, creating a sustainable foundation for continuous community engagement.

Integration with View Agents for Real-Time Feedback

Additionally, the system can integrate with View Agents to enhance transparency and feedback, including:

- Action Logging:
  LLMs track whether their suggestions were accepted or ignored, using this data to refine future guidance and improve prediction accuracy.

- User Dashboards:
  Users can access a "Suggestion Log" that displays all LLM-generated prompts they have received, along with the actions they took in response, providing clear insights into their contribution history.

Through this context-adaptive approach, users are empowered to make high-impact decisions aligned with their structural roles and community needs, fostering a more self-sustaining cycle of meaningful contributions.

(5) Structural Analysis of Behavioral Events in Virtual Spaces

User behaviors within the metaverse, such as movement, interactions, and prolonged presence, are modeled as spatial graph structures that integrate both spatial and temporal information. In this model:

- Nodes: Represent individual users.

- Edges: Represent weighted connections based on relative proximity (distance) and interaction intensity (communication volume) between users.

This approach goes beyond simple log data, revealing the complex spatial and social interaction networks that form the underlying structure of virtual spaces.

GNN-Based Detection of Structural Roles and Contribution Quantification

By applying Graph Neural Networks (GNNs) to this spatial graph structure, it is possible to automatically detect various structural roles and quantify the contribution value of different user behaviors, including:

- Spatial Hubs:
  Users who occupy central locations where many others gather, actively engaging with a large number of participants.
- Supportive Interveners:
  Users who proactively approach and engage with isolated users, bridging gaps between otherwise disconnected groups.
- Area Anchors:
  Users who remain in specific areas for extended periods, encouraging the activity of others through their persistent presence.

Quantifying Spatial Contributions

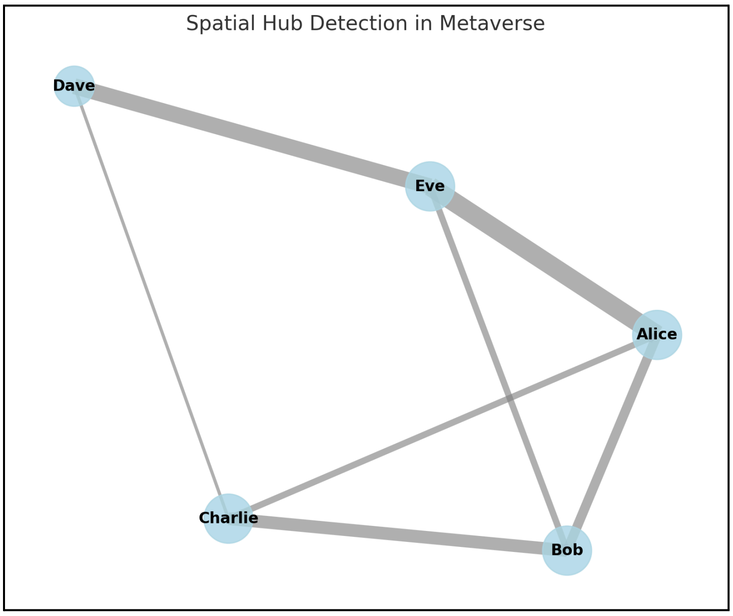
By defining and accumulating these structural behavior patterns as contribution scores, the system can capture spatial contributions that are often overlooked in traditional visualizations. This enables a more precise assessment of the impact these behaviors have on overall community dynamics and virtual space vitality.

An example of Spatial Hub Detection is presented below.

This graph represents the results of Spatial Hub Detection based on node centrality and interaction volume.

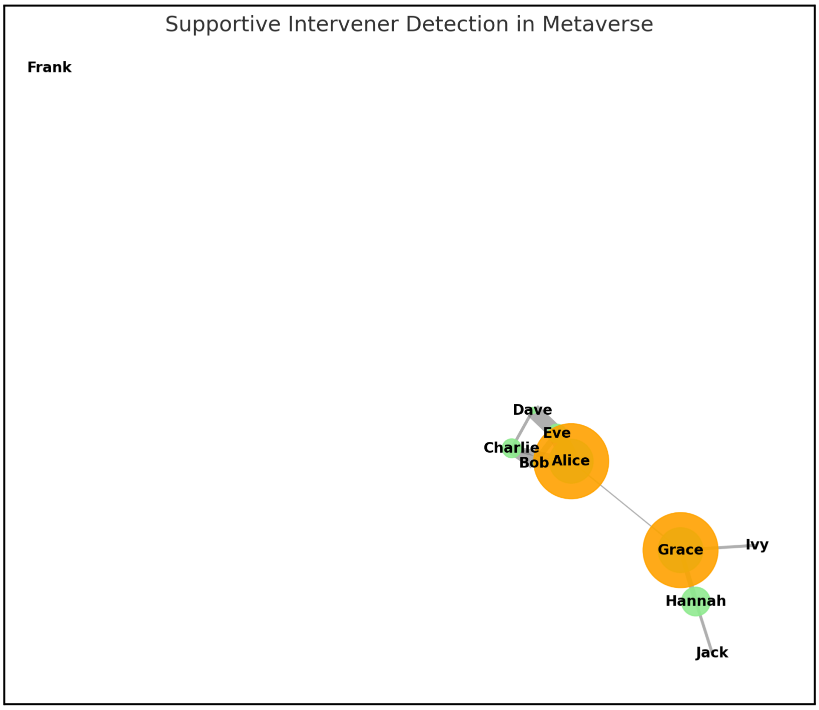
In this visualization:

- Node Size: Reflects the centrality of each user, indicating the number of connections they have with other users.
- Edge Thickness: Represents the volume of interactions (communication intensity) between connected users.



From this graph, it is clear that nodes with larger sizes and thicker edges function as "Spatial Hubs." Notably, "Eve"stands out as a central figure, actively engaging with many other users and playing a critical role in the overall network structure.

Next, the analysis results for "Supportive Interveners" are presented below.

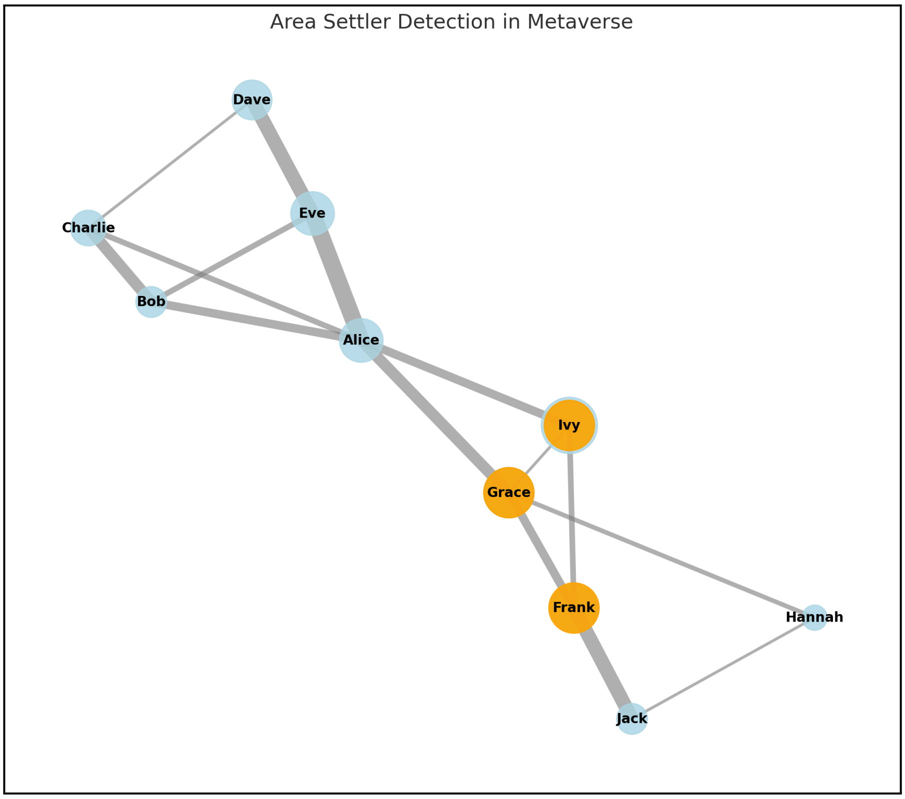


In this graph, the orange nodes represent Supportive Interveners like Alice and Grace, who proactively reach out to isolated users, increasing their connections within the network.

- Node Size: Reflects Betweenness Centrality, indicating the importance of each node in bridging connections between other nodes.
- Edge Thickness: Represents the volume of interactions between users.

These supportive interveners can be seen actively engaging with more isolated nodes, such as Hannah and Jack, demonstrating their critical role in integrating disconnected users into the broader network.

Finally, the analysis results for "Area Anchors" — users who remain in specific areas for extended periods and promote the activity of others — are presented below.

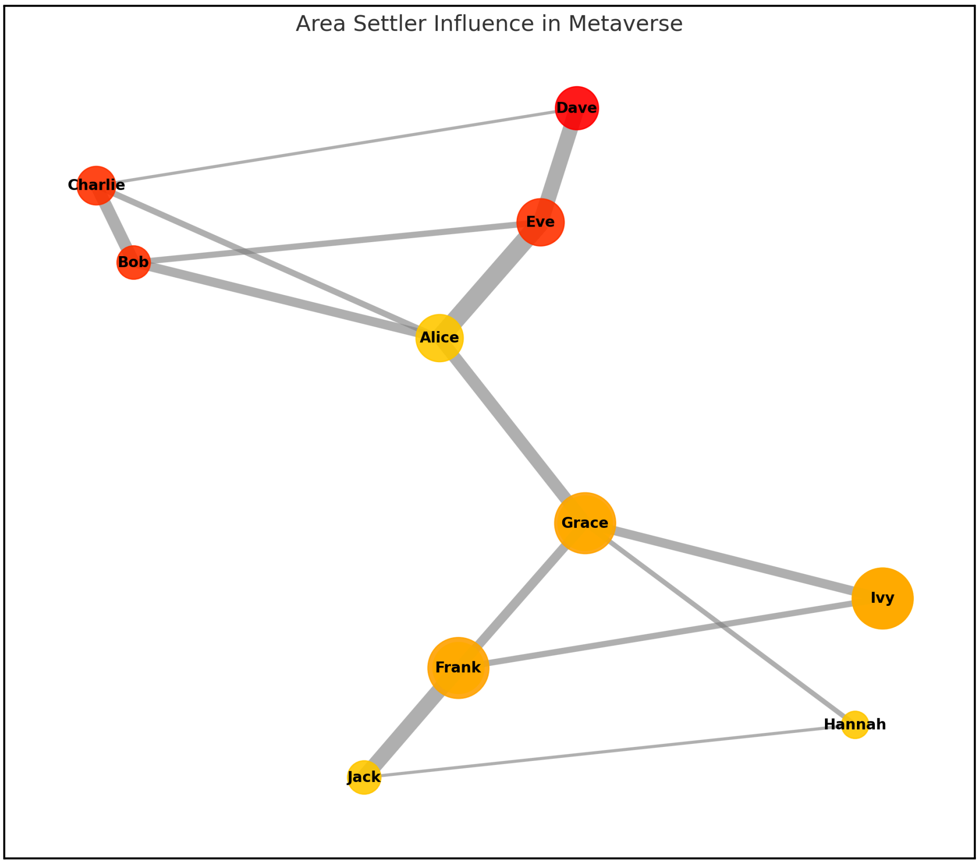


In this graph, the orange nodes represent "Area Anchors" — users like Ivy, Grace, and Frank who remain in specific areas for extended periods, encouraging the activity of others around them.

- Node Size: Reflects the duration of time each user spends in a particular area.
- Edge Thickness: Represents the volume of interactions between connected users.

Nodes with longer residence times and higher connection volumes are identified as Area Anchors, as their prolonged presence naturally attracts other users, contributing to local space activation and community formation.

Additionally, the following analysis quantifies the influence range of these Area Anchors and the activation effects they have on surrounding users.



In this graph, the orange nodes represent Area Anchors (Ivy, Grace, Frank). The visualization elements are defined as follows:

- Node Color Intensity: Indicates the influence range from each Area Anchor, with intensity decreasing as distance increases.
- Node Size: Reflects the duration of time each user has spent in a particular area.
- Edge Thickness: Represents the volume of interactions between connected users.

Key Insights from the Analysis

- Nodes closer to Area Anchors tend to have darker colors, indicating higher levels of activation and influence.
- Ivy and Grace, in particular, attract a large number of nearby users, demonstrating extensive influence ranges.
- These results clearly illustrate the spatial impact of long-residence users in promoting local activity and community formation.

Integration with View Agents for Advanced Visualization

When integrated with View Agents, this analysis enables additional advanced visualizations, including:

- Heatmap Display: Visualize where users gather and identify central figures over different time periods.
- Interaction Network Diagrams: Map user connections based on interaction intensity.
- Role-Labeled Node Diagrams: Add GNN-generated role labels like "Hub," "Supporter," and "Anchor" to each node.
- Contribution Timeline: Plot the changing roles and influence of users over time.

In this way, structurally analyzing spatial × interaction behavior logs with GNNs provides a clear, quantitative assessment of user contributions and area activation, making it possible to evaluate both the quality of interactions and the broader impact on the virtual environment.

(6) Integration with Visualization and Feedback (View Integration)

The View Agent plays a crucial role in visualizing user actions, messages, and dynamic network structures, presenting analytical results in an intuitive and easily understandable format. Specifically, it can map how users engage with different topics, assess the impact of their messages on other users, and visualize the shifting roles of individuals within the network over time (e.g., from peripheral to central positions).

Additionally, by linking concrete evidence of user contributions (e.g., message content, reactions, thread branches) with their structural positions in the network, the system provides a unified view of contribution behavior and structural influence. This allows both users and community managers to clearly understand changes in influence and growth over time, providing a solid foundation for feedback and incentive design.

Examples of View-Based Visualizations

1. Topic Influence Analysis:
   If User A posts an opinion in Topic A, which then sparks responses from multiple users and leads to a related discussion in Topic B, the View Agent can visualize this Topic A → Topic B propagation path using arrows, highlighting how User A's initial post served as a trigger for broader topic generation.
2. Network Role Transition:
   If User A starts as a peripheral node with only 1-2 connections but becomes more active over a few weeks, leading to more connections and a shift toward the network center, the View Agent can display this transition through a time slider, showing the node's movement from "Peripheral" to "Central" status along with a corresponding increase in Degree Centrality.
3. Contribution Evidence and Structural Position:
   If User A initiates a discussion by raising a critical issue (e.g., "There is an ethical concern here"), which then attracts empathy and further discussion from other users, eventually leading to actionable proposals (e.g., "Here is a potential solution"), the View Agent can label each message (e.g., Issue Identification / Empathy Formation / Solution Proposal) and map these messages to their respective nodes within the network.
4. Feedback and Incentive Integration:
   If User A achieves a significant Influence Score increase, earning an NFT badge such as "Discussion Catalyst,"the View Agent can display this alongside a score progression graph, clearly illustrating the relationship between actions and rewards.

Cross-Reality Agent (LLM + GNN Integration)

The Cross-Reality Agent integrates "Meaning Understanding" (what type of contribution, as determined by LLMs) and "Structural Analysis" (where and how the contribution impacted the network, as determined by GNNs) to evaluate and utilize user actions as socially meaningful contributions within virtual environments.

This combination enables not just basic point-based rewards, but a more nuanced engagement design that leverages social structures for more meaningful and context-aware reward systems, creating a solid foundation for sustained, purpose-driven community interactions.

3.2.6. View Agent

The View Agent is an interface agent responsible for visualizing, analyzing, and explaining the operational results of the multi-agent system, including user contribution histories, score progressions, and reward details. It provides real-time situational awareness and decision-making support not only to individual users but also to administrators and strategy designers, enabling more effective community management and intervention.

(1) User-Specific Contribution Visualization

This system allows for the visualization of each user's contribution scores (e.g., Behavior, Knowledge, Diffusion, Economic) over time, which are stored as time-series data in XTDB. In addition to raw score progressions, the system also incorporates behavioral labels generated by LLMs (Large Language Models) (e.g., "Answered FAQ", "Provided Support to Others") and structural roles within the network identified by GNNs (Graph Neural Networks) (e.g., "Information Broadcaster", "Hub", "Bridge"). This comprehensive view provides a more intuitive understanding of social and knowledge contributions that might be difficult to capture through numerical scores alone.

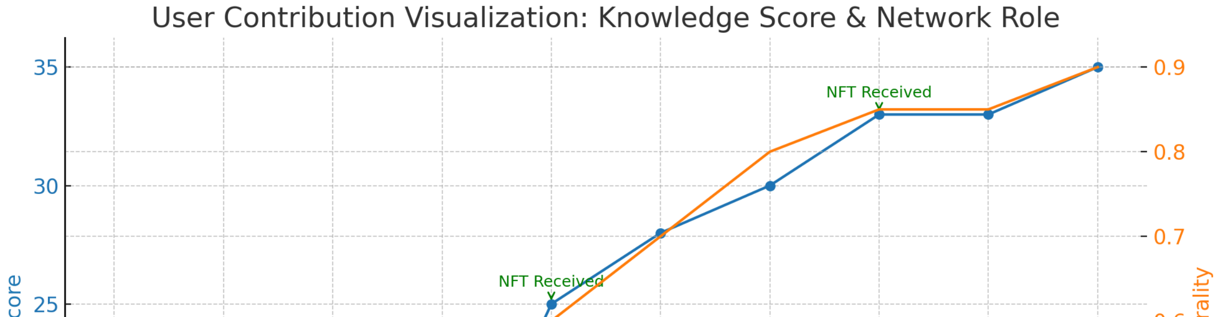
For example, the system can clearly highlight relationships between actions, influence, and timing, such as:

- "Knowledge score surged on April 10, 2025 (answered 3 FAQs)"
- "Network centrality score increased around the same time, indicating a shift to a hub role"

Additionally, NFT and token reward issuance histories can be displayed on the same timeline, allowing users and administrators to visually track the causal relationships between specific contributions and the rewards they generated. This makes it possible to clearly understand "what contributions led to what returns," providing actionable insights for personalized feedback and strategy design.

This type of contribution visualization framework supports the creation of highly transparent evaluation and reward models, enhancing user motivation and promoting sustained engagement within the community.

Below are some specific examples of how this can be implemented.



The above diagram is a specific example of User Contribution Visualization, simultaneously displaying "Knowledge Score Time-Series" and "Network Centrality Score (Role Evaluation by GNN)" for a single user.

- Blue Line: Knowledge Score (e.g., reflecting the number of FAQs answered)
- Orange Line: Network Centrality (structural influence within the network)
- Green Arrows: Timing of NFT rewards received

For example, on April 5, 2025, there is a sharp increase in the Knowledge Score (likely due to multiple FAQ responses), and an NFT reward was issued on the same day, clearly visualizing the causal relationship between contribution and reward. Additionally, the Network Centrality Score also increased, suggesting that the user transitioned into a hub-like position within the network during this period.

This type of visualization not only enhances the explainability of contribution evaluations but also serves as a valuable tool for user feedback and the optimization of community strategies.

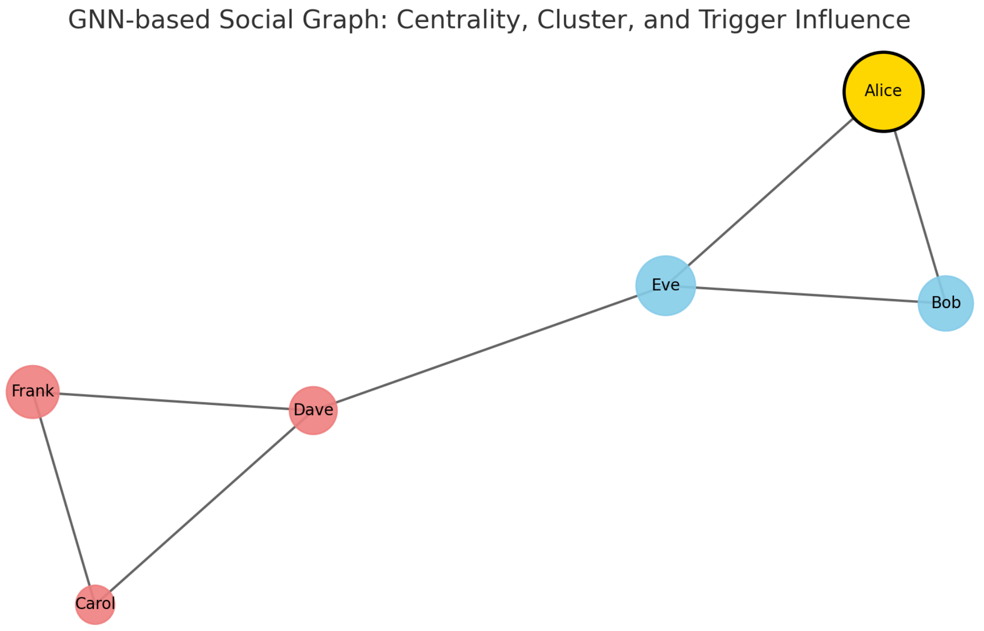
(2) Network Structure and Diffusion Analysis (GNN Integration)

In this system, GNN (Graph Neural Network) Agents analyze the social graph formed by user interactions, including reply chains, co-occurrence of messages, and citation structures, to quantitatively assess the overall network structure and the influence of individual users.

For visualization, each user is represented as a node, with the following features:

- Node Size: Reflects Centrality Scores, indicating the influence and importance of each user within the network.
- Node Color: Represents cluster membership, allowing for quick identification of community affiliations.
- Visual Emphasis: Nodes with high influence are highlighted for greater visibility, making it easier to identify key figures within the network.

Additionally, the system can highlight "Trigger Messages" — specific posts that sparked significant user responses or triggered widespread diffusion. These trigger points can be annotated and visually emphasized to clearly illustrate the origins and pathways of influence within the network.



GNN-based Social Graph: Centrality, Cluster, and Trigger Influence

This network-based approach to diffusion analysis provides deep insights into "who spread what information, how, and at what point they influenced collective behavior." This structural understanding supports more strategic decision-making and targeted intervention in community management.

The diagram above is a concrete example of a Social Graph visualization, analyzed by the GNN Agent to capture the complex network of replies, co-occurrences, and message interactions between users.

Explanation of the Visualization:

- Node Colors:
  - Sky Blue: Cluster 1 (e.g., Technical Discussion Cluster)
  - Light Coral: Cluster 2 (e.g., Casual Chat and Support Cluster)
- Node Size:
  - Proportional to each user's Centrality Score, reflecting their importance within the network structure. Larger nodes indicate higher influence.
- Highlighted Nodes (Thick Border + Gold Color):
  - Alice is highlighted as a "Trigger User," indicating that her posts have induced significant reactions from other users, acting as a key catalyst for information spread.

- Edges (Lines):
    - Represent the reply, co-occurrence, and interaction relationships between users. Cross-cluster connections (e.g., between Eve and Dave) are also visualized, providing insight into which users act as bridges for information flow between different groups.

By combining this type of graph with contribution score progressions and message histories, it becomes possible to visually identify "which messages had the most influence" and "which users played critical roles in information diffusion," providing deeper insights into the social dynamics of the network.

(3) Cross-Reality Contribution Map

This system provides a Cross-Reality Contribution Map to capture and integrate user contributions across both real-world and metaverse environments. This map is powered by Cross-Reality Agents that aggregate activity logs (e.g., real-world event participation, virtual support, guidance, and interactions) into a unified, chronological timeline, allowing for user-level or event-level contribution history visualization.

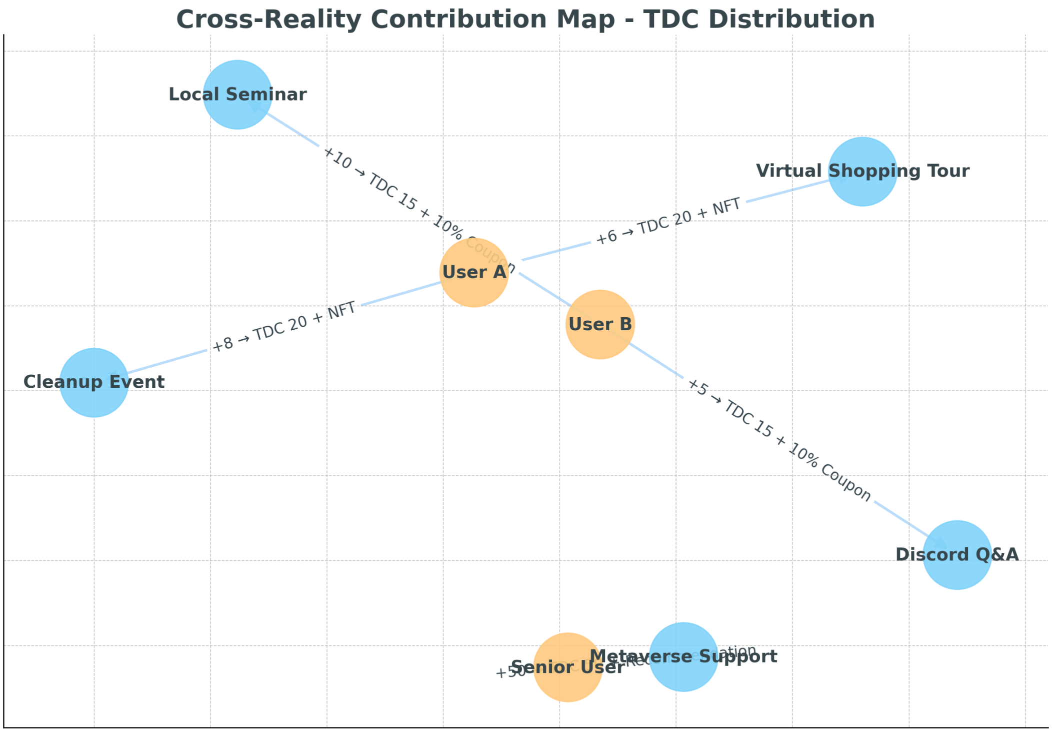
For example, if a user participates in a "morning community cleanup in the physical world" and then "guides new participants in the metaverse later that evening," these actions are seamlessly displayed on a single, continuous contribution timeline. This approach overcomes the fragmentation of traditional evaluation models by providing a fair and integrated view of contributions across physical and virtual spaces.

Key Use Cases for the Cross-Reality Contribution Map:

1. User-Level View:
    - Provides a time-series, space-specific overview of each user's contributions (e.g., 3 real-world activities, 5 metaverse engagements), allowing for a comprehensive understanding of individual impact.
2. Event-Level View:
    - Offers a comprehensive view of who did what, where, and when during specific events (e.g., local festivals, online study sessions), providing complete coverage of participant contributions.
3. Feedback and Reward Integration:
    - Enables the design of reward systems based on the type and location of contributions (e.g., bonus TDC distribution for combining real-world activities and virtual support).

This unified cross-reality view is more than just an activity log; it provides a 360-degree perspective on user contributions, serving as a foundational tool for building empathy, trust, and incentive structures within hybrid communities.

Below is a sample Cross-Reality Contribution Map UI for reference.



This diagram represents the Cross-Reality Contribution Map, which visualizes user contributions in both real-worldand metaverse environments, along with their corresponding TDC (local currency) and NFT rewards. By clearly linking user actions with their resulting rewards, this map provides an intuitive understanding of the cause-and-effect relationship between contributions and rewards.

Node Types (Circles)

- Users (Orange):
  - Represents users who are active in both real and virtual spaces (e.g., User A, User B, Senior User).

- Events (Light Blue):
  - Represents specific contribution activities in the real world and the metaverse (e.g., Cleanup Event, Virtual Shopping Tour, Local Seminar, Discord Q&A, Metaverse Support).

Edges (Arrows)

- Show the connections between users and the events they participated in, along with their corresponding scores and rewards.
- For example, "+8 → TDC 20 + NFT" or "+50 → TDC 50 + Recommendation" clearly indicate the contribution scores and the rewards received, making it easy to see the impact of each action.

Key Use Cases for This Dashboard

1. Improving Contribution Transparency:
   - Clearly displays how each user earned their scores and received rewards, ensuring transparency in reward distribution.
2. Enhancing User Motivation:
   - Visually demonstrates how contributions are evaluated and what incentives they lead to, encouraging further user engagement.
3. Evaluating Policy Impact:
   - Provides quantitative insights into how real-world and metaverse activities are connected and contribute to regional revitalization.
4. Supporting Cross-Reality Strategy Design:
   - Visualizing the distribution of scores and rewards based on real-world behavior data can inform the design and improvement of future initiatives.

Broader Impact and Future Potential

By leveraging Cross-Reality Contribution Maps, it becomes possible to design sustainable participation models that transcend physical and digital boundaries, supporting the formation of decentralized social capital. This approach facilitates multi-region and multi-layer collaborations, promoting long-term engagement across diverse communities.

(4) Feedback Visualization for Incentive Design

This system includes a feedback visualization mechanism that clearly explains the evaluation logic behind incentives(e.g., local currencies, NFTs) provided to users. This feature, powered by the collaboration between the Metric Analyzer Agent and LLM, aims to present the causal relationship between score items and rewards in a way that is easily understandable for users.

For example, if a user has consistently provided technical support over several weeks, resulting in their Technical Support Score surpassing a predefined threshold, leading to the issuance of TDC + NFT, the UI might display an explanation like:

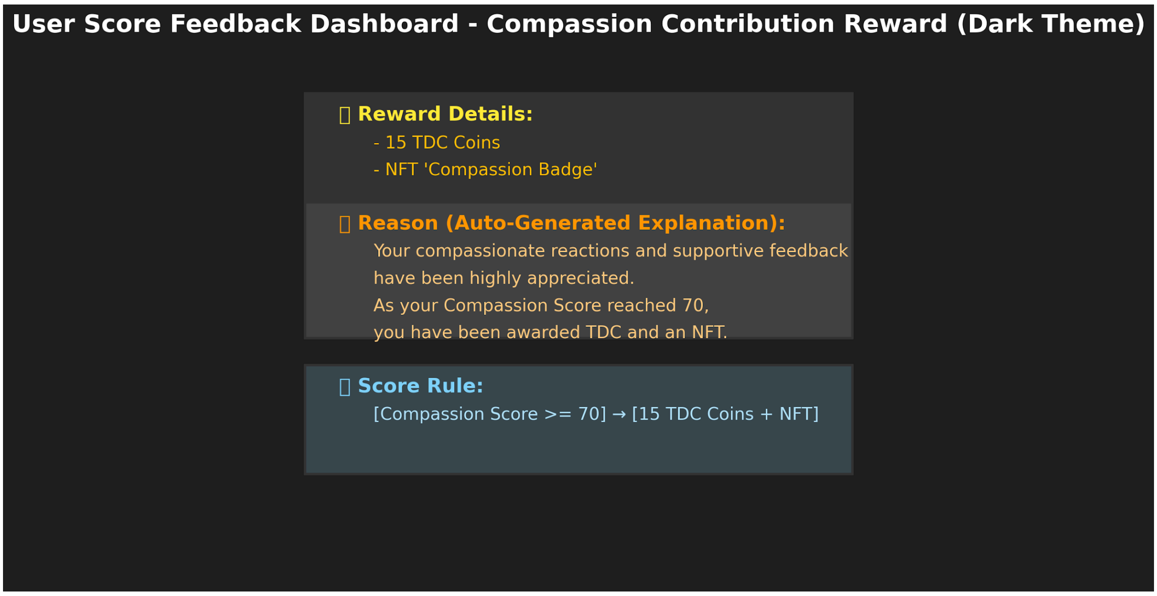
"Over the past 30 days, you have answered more than 10 technical questions on Discord and in the metaverse, resulting in your Technical Support Score reaching 85. As a result, you have been awarded this month's Community Support NFT."

This explanation is structured based on score changes and rule application results generated by the Metric Analyzer, while the LLM Agent converts this technical data into natural, user-friendly language, minimizing the use of complex terminology.

Key Features of the Feedback System:

- Rule Transparency:
    - Users can also view the underlying reward rules themselves (e.g., "NFT awarded for Technical Support Score 80 or higher"), ensuring a high level of transparency in the incentive design.
- Feedback History:
    - This feedback is stored on the View Agent dashboard as a Feedback History, allowing users to review the relationship between their contribution history and rewards over time.

Below is an example of a Feedback UI for Empathy Rewards for reference.



User Score Feedback Dashboard - Compassion Contribution Reward (Dark Theme)

▯ **Reward Details:**

- 15 TDC Coins
- NFT 'Compassion Badge'

▯ **Reason (Auto-Generated Explanation):**

Your compassionate reactions and supportive feedback
have been highly appreciated.
As your Compassion Score reached 70,
you have been awarded TDC and an NFT.

▯ **Score Rule:**

[Compassion Score >= 70] → [15 TDC Coins + NFT]

User Score Feedback UI for Empathy-Based Contribution Rewards

The diagram illustrates the user score feedback UI for empathy-based contribution rewards. It is structured into the following three main sections:

1. Reward Details

This section displays the specific rewards earned by the user.
Examples:

- "15 TDC Coins"
- "Compassion Badge (NFT)"
  These rewards are clearly outlined based on the user's score.

2. Reason - Auto-Generated Explanation

This part provides an automated explanation of why the rewards were granted.
It generates messages based on the user's contribution actions, such as empathetic reactions or feedback posts.
Examples:

- "You received 15 TDC Coins for actively engaging in community discussions."
- "You earned a Compassion Badge for consistently supporting others with thoughtful feedback."

3. Score Rule

This section briefly summarizes the scoring rules that serve as the basis for rewards.
Examples:

- "15 TDC Coins + NFT awarded for achieving a compassion score of 70 or higher."
- "10 TDC Coins for 10 positive reactions within 30 days."
  This helps users clearly understand the thresholds required to unlock specific rewards.

Purpose of the Dashboard

The dashboard serves the following key purposes:

- Enhancing User Engagement: Provides clear goals, encouraging proactive contributions.

- Improving Transparency: Clearly communicates the reasons for reward distribution, enhancing trust and transparency.
- Strengthening Behavioral Feedback: Allows users to intuitively understand which actions are valued, promoting sustained contributions.
- Maintaining Motivation: Provides clear feedback, helping users maintain and boost their motivation.

Benefits for Administrators and System Operators

- Accountability: Provides clear evidence to address user questions like, "Why did I receive this reward?" or "Why was I not eligible?"
- Reproducibility: Clearly outlines the cause-and-effect relationship between actions and rewards, making it easier for users to plan future contributions.
- Strategic Planning: Allows for the design of new reward rules and behavior encouragement strategies based on past score and reward data.

By providing this clear "Score → Reward" causal feedback through visualization, automated explanation generation, and activity log tracking, the system aims to improve the transparency of TDC issuance while simultaneously maintaining user motivation and ensuring system reliability.

(5) Macro-Level Effect Visualization (Macro Analysis)

This system provides not only user-level contribution score analysis but also macro-level visualization of the overall impact of initiatives within a region. Specifically, it supports the statistical display and aggregation of key economic indicators such as:

- Total Contribution Score Trends: Regional and time-based tracking of overall contribution scores.
- Total Issuance of Local Currency (TDC) and NFTs: Cumulative data on the total amount of TDC and NFTs distributed as rewards.
- Currency Circulation Volume: Analysis of the total volume of local currency in circulation.
- Return Visit Rates to Events and Facilities: Metrics that reflect the economic impact through repeat visits and participant engagement.

These metrics are tightly integrated with the KPIs (Key Performance Indicators) set for each initiative, enabling the comparison of "Planned vs. Actual" values, as well as visual representation of achievement rates. This facilitates the evaluation of initiative outcomes and the establishment of continuous improvement cycles.

Additionally, the system includes an automated geolocation-based heatmap generation feature, allowing for intuitive understanding of participation trends and contribution distribution across different areas. This capability can be leveraged for correcting regional disparities and planning targeted interventions.

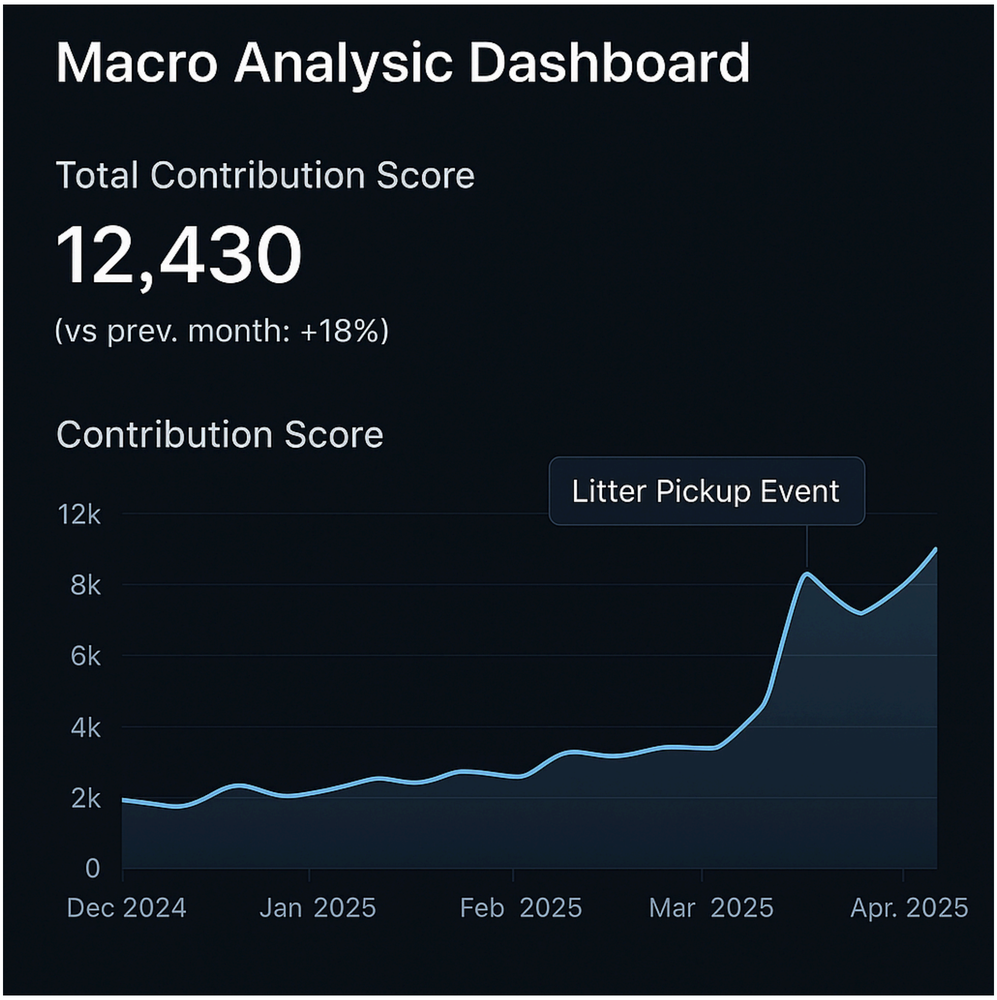
Example Use Cases:

- "Total Contribution Score for City X in April 2025 increased by 18% compared to the previous month."
- "Total NFT Issuance reached 220 units."
- "Return Visit Rate is 65%, up 12 points from the same month last year."

These data points can be simultaneously displayed on the dashboard through line charts, bar graphs, and heatmaps, providing a comprehensive overview for local governments, regional organizations, and corporate partners, supporting collective decision-making for future initiative planning.

In this way, a quantitative evaluation mechanism that provides a macro-level view of the entire region serves as a foundational system for sustainable improvement of community engagement programs centered around local currency and the construction of resident-driven regional economies.

Below is an example of a macro analysis UI for this purpose.



Macro Analysic Dashboard

Total Contribution Score

12,430

(vs prev. month: +18%)

Contribution Score

Visualization of Total Contribution Scores for a Community Participation Project Using Local Currency in a City

This dashboard provides a quantitative visualization of the results of a community participation promotion project using local currency introduced by a specific city. The main components of this dashboard are as follows:

Total Contribution Score

- Main Display: "Total Contribution Score 12,430 (vs prev. month: +18%)"
- Overview: This section aggregates the cumulative contribution score for the entire project on a monthly basis, displaying the percentage change from the previous month. As of April 2025, the total cumulative score stands at 12,430 points, reflecting an 18% increase compared to the previous month.

Contribution Score Trends

- Graph Format: A line chart displaying the contribution scores from December 2024 to April 2025.
- Key Features:
    - A significant spike in contribution scores is observed following the "Litter Pickup Event," indicating a temporary surge in project participation.
    - This type of visualization helps identify the impact of specific events, providing a basis for evaluating the reproducibility of successful initiatives.

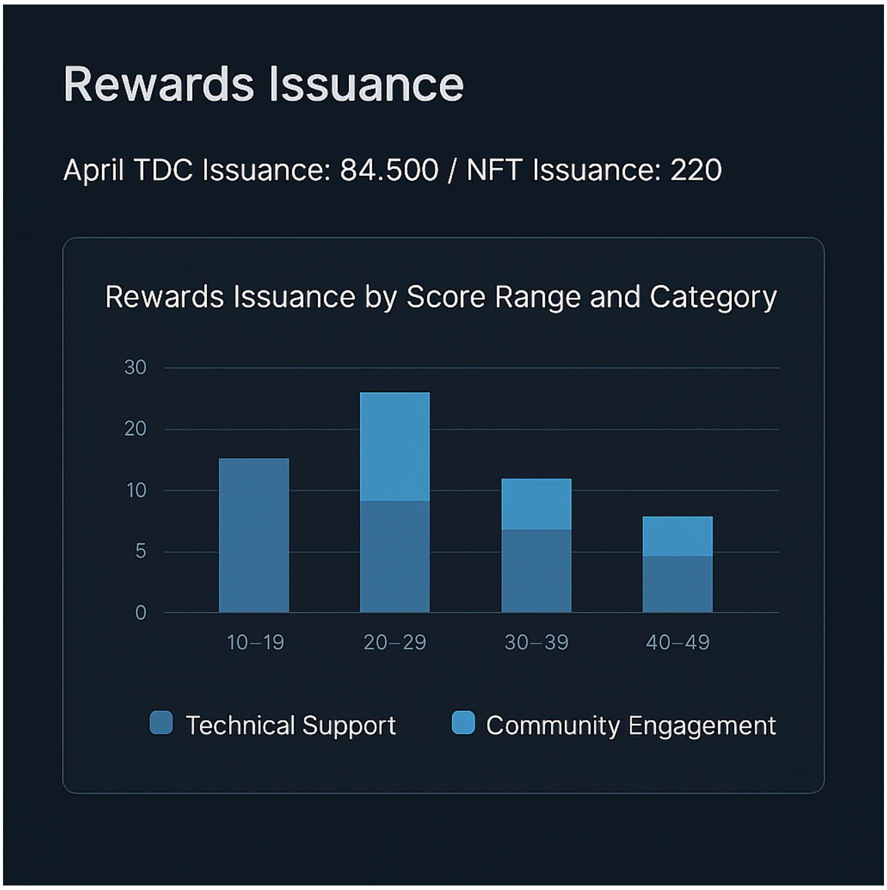
Dashboard Objectives

This dashboard is utilized for the following purposes:

1. Evaluating the Impact of Initiatives:
   Quantifies the influence of specific initiatives on citizen participation, serving as evidence for scaling successful programs to other regions or themes.
2. Strategy Improvement and Expansion Decisions:
   Identifies high-impact, short-term initiatives like the "Litter Pickup Event" and optimizes their scale and scope for future iterations.
3. Enhancing Citizen Engagement:
   Analyzes changes in citizen interest and motivation based on contribution score fluctuations, supporting the development of initiatives that boost participation.

4. Promoting Local Currency Circulation:
Examines the correlation between contribution scores and local currency usage, fostering long-term economic activation and increased participation rates.

Next, an example UI for visualizing reward (TDC/NFT) issuance will be presented.



Rewards Issuance

April TDC Issuance: 84.500 / NFT Issuance: 220

Rewards Issuance by Score Range and Category

Dashboard for Quantitative Monitoring of Local Currency (TDC) and NFT Issuance in a Community Participation Project (2025 Onwards)

This dashboard is designed to provide a quantitative overview of the issuance of local currency (TDC) and NFTs in a resident participation promotion project launched by a city in 2025. The main components are as follows:

Total Issuance

- Main Display: "April TDC Issuance: 84,500 / NFT Issuance: 220"
- Overview: This section presents the total monthly issuance of TDC and NFTs, providing a comprehensive view of the incentives distributed for citizen contributions. For the month of April, 84,500 TDC coins and 220 NFTs were issued, reflecting the overall scale of the project.

Rewards Issuance by Score Range and Category

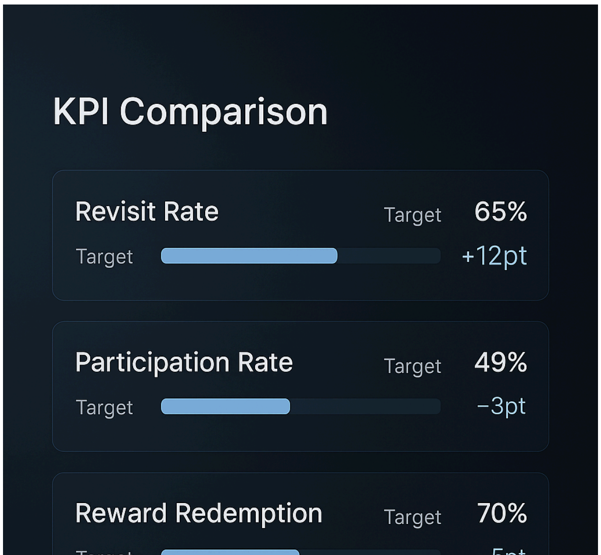
- Bar Chart Format: This component visualizes the distribution of TDC and NFT issuances by score range (e.g., 10-19, 20-29) and category (e.g., Technical Support, Community Engagement).
- Key Features:
    - Consists of two main categories: Technical Support and Community Engagement.
    - Allows for a clear comparison of issuance volumes across different score ranges.
    - The 20-29 score range shows the highest volume, indicating that technical support is the primary contribution category.

Dashboard Objectives

This dashboard serves the following purposes:

1. Evaluating Initiative Effectiveness:
   Analyzes the distribution of rewards across different score ranges and categories, providing insights into citizen interests and behavioral trends for future initiative planning.
2. Optimizing Incentive Design:
   Balances contributions across various categories, such as technical support and community engagement, to refine the incentive system.
3. Promoting Citizen Participation:
   Visualizes which score ranges receive the most rewards, motivating participants to achieve higher scores and encouraging repeat engagement.
4. Ensuring Transparency and Trust:
   Provides quantitative records of reward issuance, enhancing system credibility and strengthening relationships between local governments and citizens through transparent data sharing.

An example of a KPI comparison review dashboard is shown below.



KPI Comparison

Revisit Rate Target 65%
Target +12pt

Participation Rate Target 49%
Target −3pt

Reward Redemption Target 70%

Dashboard for Quantitative Monitoring of Key Performance Indicators (KPIs) in a Community Participation Project (2025 Onwards)

This dashboard provides a quantitative visualization of the goal achievement status for the main KPIs (Key Performance Indicators) in a resident participation promotion project launched by a city in 2025. The main components are as follows:

Progress Bars for Each KPI

- Revisit Rate
  - Target: 60%
  - Actual: 65% (+12pt)
  - Result: Achieved 12 points above the target.
- Participation Rate
  - Target: 50%
  - Actual: 49% (-3pt)
  - Result: Slightly below target.
- Reward Redemption
  - Target: 75%
  - Actual: 70% (-5pt)
  - Result: Below target, indicating room for improvement.
- Task Completion
  - Target: 80%
  - Actual: 78% (-2pt)
  - Result: Close to the target, but slightly underperforming.

Visual Progress Indicators

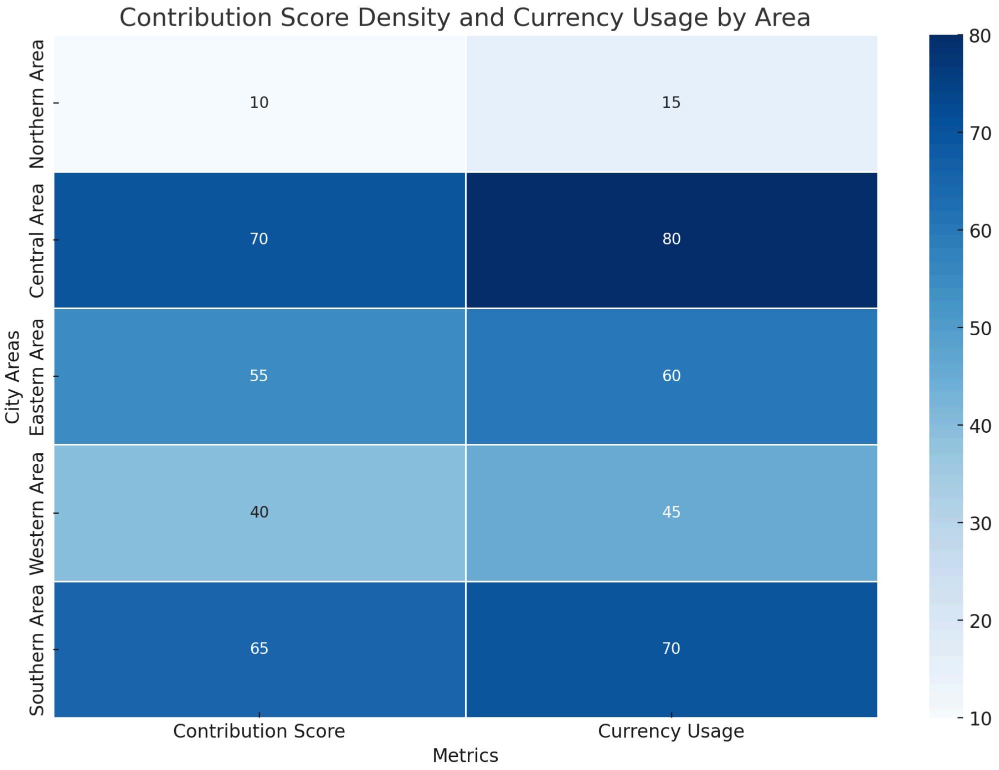
- Each KPI is represented with a blue progress bar, making it easy to identify performance gaps at a glance.
- Positive and negative differences are visually emphasized to provide intuitive feedback on achievement status.

Dashboard Objectives

This dashboard is utilized for the following purposes:

1. Measuring Initiative Effectiveness:
    - Enables real-time monitoring of goal achievement for each KPI, supporting the identification of successful initiatives and areas for improvement.
2. Visualizing Project Outcomes:
    - Provides a clear view of critical behavioral indicators like participation rates and revisit rates, serving as a valuable resource for stakeholder reporting.
3. Strategy Development and Improvement:
    - Facilitates root cause analysis for underperforming KPIs (e.g., Reward Redemption) and informs the design of future initiatives.
4. Promoting Continuous Participation:
    - Encourages sustained resident engagement by improving revisit rates and task completion metrics.

An example of a UI for area-based heatmaps is shown below.



Heatmap of Contribution Score Density and Currency Usage by Area in a City

This diagram is a heatmap illustrating the "Contribution Score Density" and "Currency Usage Count" for each area (North, Central, East, West, and South) within a city. It quantitatively visualizes the levels of resident contributions and local currency (TDC) usage in each region, with color intensity representing the relative density of scores and transaction volumes. Darker colors indicate higher contribution scores and more frequent currency usage, while lighter colors indicate lower levels.

Dashboard Objectives

This heatmap is utilized for the following purposes:

1. Decision Support for Initiative Improvement:
    - Areas with lower contribution scores (e.g., North) may have lower revisit rates, suggesting the need for more targeted event locations or support measures to boost participation and economic impact.
2. Development of Area-Specific Strategies:
    - Highly active areas (e.g., Central and South) can provide insights into successful factors, which can be replicated in less active areas to drive broader community engagement.
3. Quantitative Evaluation of Initiative Impact:
    - Regular heatmap updates enable quantitative tracking of initiative progress and effectiveness, supporting data-driven strategic decision-making.

Strategic Value of Macro Analysis

Macro analysis serves as a core function for quantitatively demonstrating the outcomes of resident participation initiatives centered around TDC, supporting the following key goals:

- Improved Policy Decision-Making:
    - Enables data-driven decisions based on statistical evidence, rather than solely relying on subjective feedback or resident sentiment, facilitating more precise targeting of initiatives.
- Faster PDCA Cycle for Initiatives:
    - Allows for immediate analysis of underperforming KPIs and rapid feedback into the planning of next-stage initiatives, improving the overall cycle time of policy adjustments.
- Visible Return on Investment (ROI):

    - Facilitates the calculation of metrics like "Revisit Rate Increase per TDC Issued," providing clear ROI evaluations for each initiative.
- Enhanced Accountability to Citizens:
    - Supports transparent reporting to residents by clearly illustrating "how the city utilized rewards and what effects they generated," strengthening trust and accountability.

This approach not only enhances the precision of policy formulation but also promotes ongoing optimization of resident participation strategies, fostering stronger collaboration between local governments, residents, and partner organizations.

(6) Dashboard Customization and Multi-Level Display

This system includes a "Customization and Multi-Level Display" feature, allowing the dashboard content, granularity, and functionality to be flexibly adjusted based on the role and usage context of each user. This capability supports UI access control for different viewer roles, such as administrators, participants, municipal staff, and regional partners, ensuring that each user receives the most relevant information and operational environment for their needs.

Role-Based Dashboard Customization

For example:

- Administrator View:
    - Access to editing score logic, reward rules, aggregate KPI management, and overall trend analysis.
    - Comprehensive controls for monitoring system performance and optimizing engagement strategies.
- General Participant View:
    - Limited to viewing personal score history, reward receipt logs, and behavior feedback.
    - Streamlined design to eliminate unnecessary complexity and focus on user engagement.

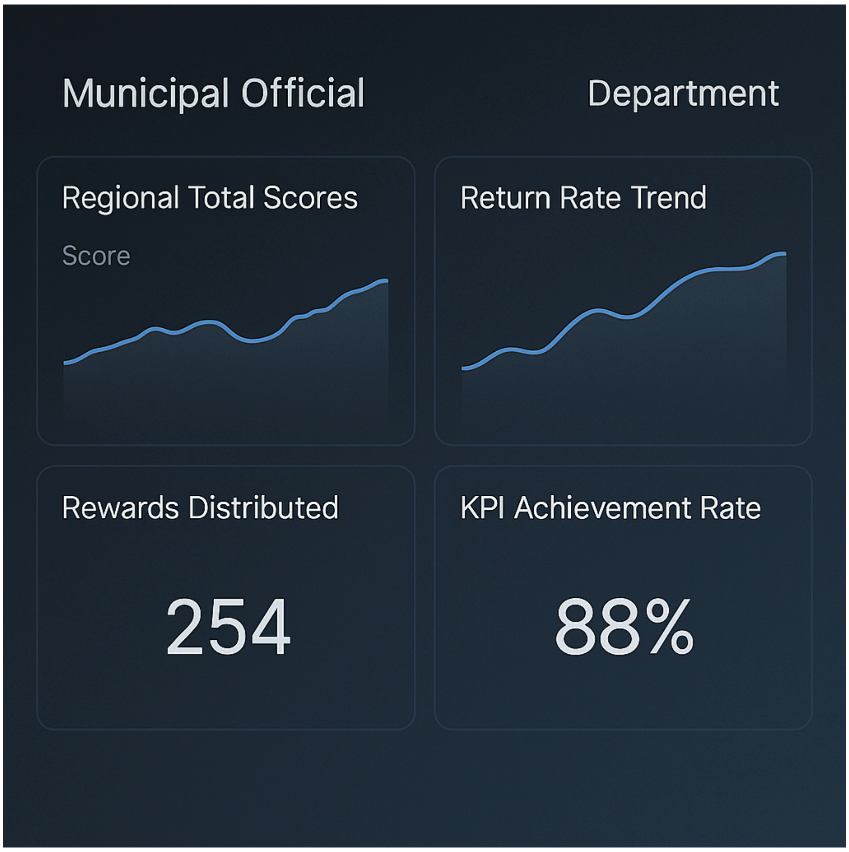
Multimodal Display for Different Environments

In addition to role-based customization, the system also supports multimodal display, optimized for various usage scenarios, including:

- Mobile Device Display:
    - Tap-first UI and vertical timeline views, allowing users to quickly check feedback and notifications.
- Web Browser Display:
    - Full-feature access, including user-level, area-level, and time-period switching for more detailed analysis.
- Large Display Mode:
    - Designed for public use in municipal lobbies or event venues, providing continuous, real-time displays of total community scores and participation heatmaps as a "public dashboard."

Benefits of Dynamic Dashboard Switching

This dynamic dashboard switching, based on user role and environment, facilitates comprehensive information sharing among all stakeholders while providing context-appropriate operability and controlled information disclosure. This approach balances security and usability, serving as a critical interface for managing both the transparency and efficiency of the overall system.



Municipal Official

Department

Regional Total Scores

Score

Return Rate Trend

Rewards Distributed

254

KPI Achievement Rate

88%

Below is an example UI for a regional currency use case, focusing first on the perspective of Municipal Policy Managers.

Dashboard for Municipal Staff (Policy Departments) – Regional Activation Performance Monitoring

This dashboard is designed to provide municipal staff, particularly in policy departments, with a quantitative overview of the outcomes of regional activation initiatives. It consists of four main indicators, each focusing on a critical aspect of community engagement and impact measurement:

Main Indicators

1. Regional Total Scores:
   - Visualizes the cumulative contribution scores by region through line graphs.
   - Tracks the total amount of scores accumulated through technical support and community contribution activities, allowing for the measurement of initiative effectiveness by region.
2. Return Rate Trend:
   - Displays the repeat participation rates for regional events and digital spaces over time.

- Serves as a key indicator of ongoing community engagement and long-term involvement.

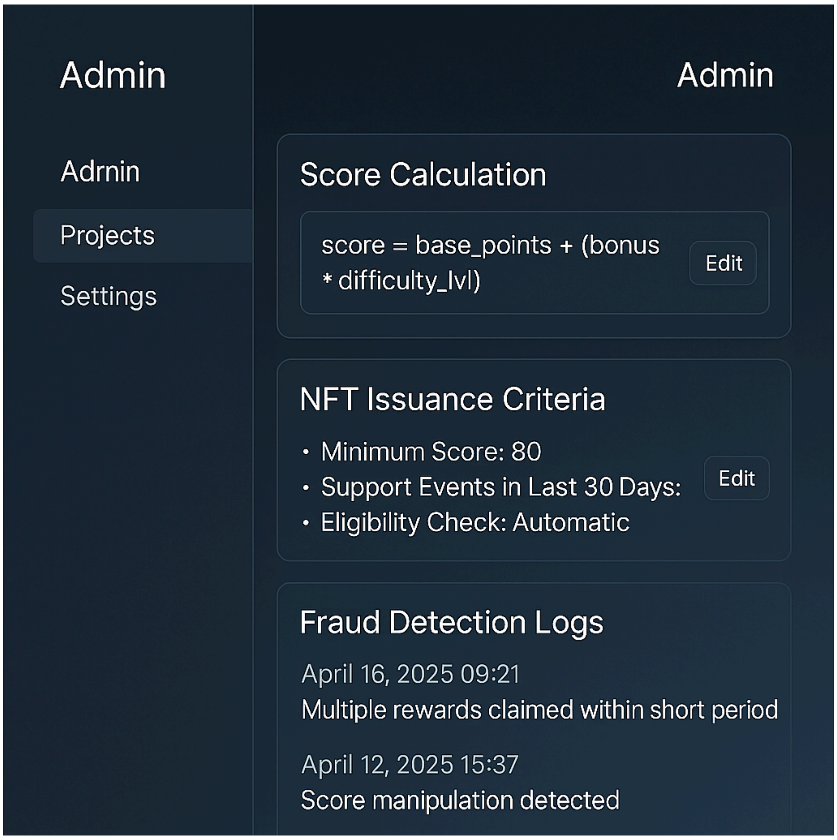
3. Rewards Distributed:
   - Clearly presents the number of rewards distributed, including NFTs and local points, providing a concrete measure of the effectiveness and cost efficiency of incentive programs.
4. KPI Achievement Rate:
   - Shows the achievement rates for pre-set policy goals (e.g., number of participants, volume of contribution actions) as a percentage, supporting strategic progress tracking.

Dashboard Use Cases

This UI is designed for the following purposes:

1. Monitoring and Improving Regional Initiatives:
   - Provides a data-driven basis for evaluating the impact of policies, enabling rational decision-making for next steps (e.g., strengthening specific initiatives or reallocating focus to different regions).
2. Internal Reporting and Communication:
   - Allows for immediate access to key performance metrics, facilitating the creation of reports and evidence-based explanations for other departments and upper management.
3. Transparency and Stakeholder Communication:
   - Enhances the credibility of incentive programs by clearly presenting reward distribution numbers and participation trends, building trust among external partners and citizens.

An example UI for Administrator View is outlined below.



Score Calculation

score = base_points + (bonus * difficulty_lvl)

NFT Issuance Criteria

- Minimum Score: 80
- Support Events in Last 30 Days:
- Eligibility Check: Automatic

Fraud Detection Logs

April 16, 2025 09:21
Multiple rewards claimed within short period

April 12, 2025 15:37
Score manipulation detected

Administrator Dashboard – Scoring System Management and Oversight

This UI screen is designed for Administrators who have the authority to design and adjust the scoring system. It primarily consists of the following three management functions:

1. Score Calculation

- Example Formula: score = base_points + (bonus * difficulty_lvl)
- Description: Allows for flexible adjustment of score calculation logic based on base points, bonus values, and difficulty coefficients.
- Edit Function: An "Edit" button enables dynamic modification of scoring rules, supporting real-time optimization in response to changing initiatives and program requirements.

2. NFT Issuance Criteria

- Minimum Score: 80
- Target Period: Past 30 days of support events
- Eligibility Check: Automatic

- Description: Clearly defines and allows for the customization of NFT issuance conditions, ensuring transparency and fairness in the reward system.

3. Fraud Detection Logs

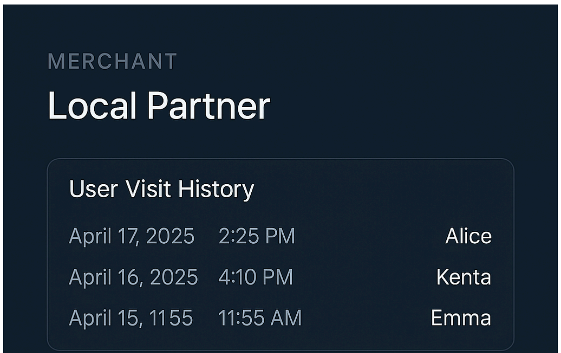
- Example Logs:
  - April 16, 2025, 09:21: Multiple reward claims detected within a short time frame.
  - April 12, 2025, 15:37: Signs of score manipulation detected.
- Description: Provides real-time visibility into suspicious activities and abnormal behavior through a log-based interface, helping maintain the integrity of the scoring and reward system.

Dashboard Use Cases

This administrator dashboard is designed for the following operational purposes:

1. Supporting System Design:
   - Enables flexible editing of score formulas and reward criteria, allowing for the creation of incentive structures that effectively drive participant behavior.
2. Ensuring System Integrity:
   - Provides real-time detection and logging of suspicious activities, maintaining the transparency and fairness of the overall reward system.
3. Strategic System Optimization:
   - Allows for dynamic adjustments to NFT issuance conditions and scoring criteria to maximize participation rates and contribution levels, optimizing overall program outcomes.
4. Supporting Reporting and Audits:
   - Automatically records score settings and fraud detection histories, facilitating data-driven decision-making and accountability in external reporting.

An example UI for Regional Partners (Stores) is outlined below.



MERCHANT

Local Partner

User Visit History

April 17, 2025 2:25 PM Alice
April 16, 2025 4:10 PM Kenta
April 15, 1155 11:55 AM Emma

Regional Partner Dashboard – Local Currency (TDC) Usage and Visitor Management

This dashboard is designed for Regional Partners (e.g., local shops, facilities, and affiliated businesses) to monitor TDC usage, visitor histories, and reward redemption status. It is organized into the following three main sections:

1. User Visit History

- Description: Displays recent visit records by date, time, and user name in chronological order.
- Use Case: Helps store owners track repeat customers and regular visitors, supporting personalized service and customer relationship management.

2. TDC Redemption History

- Description: Logs TDC usage during visits, including payment transactions and special reward redemptions.
- Use Case: Enables businesses to assess the effectiveness of TDC-based promotional strategies by analyzing usage frequency and volume.

3. Reward Redemption Status

- Description: Provides an inventory view of available rewards (e.g., coupons, free drinks) and their remaining quantities.
- Use Case: Assists in stock management, determining when to restock or adjust reward offerings for more efficient store operations.

Dashboard Use Cases

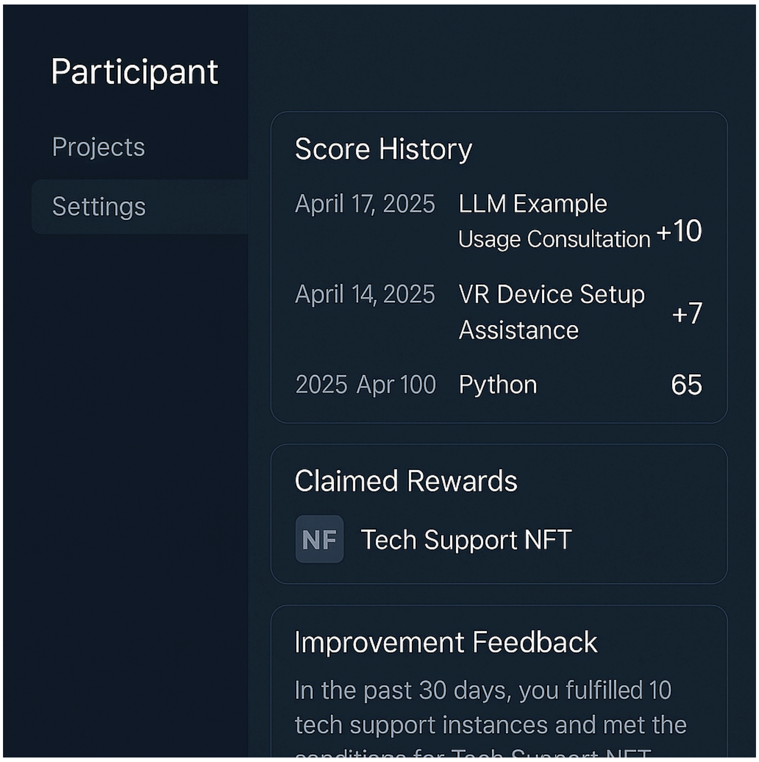
This UI is designed for the following operational purposes:

1. Visualizing Contributions to the Local Economy:
    - Provides quantitative insights into TDC usage by store, helping measure each business’s contribution to the local economic ecosystem.
2. Supporting Business Decision-Making:
    - Analyzes visit and redemption trends, guiding campaign planning and customer service improvements.
3. Optimizing Reward Supply and Demand:

- Identifies popular rewards and exchange patterns, enabling better inventory management and product lineup optimization.

4. Building Customer Relationships:
   - Uses visit history to identify frequent customers, allowing for personalized greetings, exclusive promotions, and stronger customer loyalty initiatives.

Finally, the UI screen for General Participants can be designed as follows.



Score History

April 17, 2025	LLM Example Usage Consultation	+10
April 14, 2025	VR Device Setup Assistance	+7
2025 Apr 100	Python	65

Claimed Rewards

NF Tech Support NFT

Improvement Feedback

In the past 30 days, you fulfilled 10 tech support instances and met the conditions for Tech Support NFT

Participant Dashboard – Personal Activity and Reward Tracking

This UI screen is designed for General Participants to track their own activity history and reward status. It consists of the following three main sections:

1. Score History

- Description: Displays the participant's past contribution scores, including technical support and question responses, in chronological order.
- Examples:
  - April 17, 2025: Provided support on LLM usage examples (+10)
  - April 14, 2025: Assisted with VR device setup (+7)
  - April 10, 2025: Answered a Python-related question (+5)

- Purpose: Allows participants to review their past activities, assess their overall contributions, and track personal progress.

2. Claimed Rewards

- Description: Lists the NFTs and special rewards received by the participant.
- Examples:
    - Tech Support NFT
- Purpose: Visualizes the results of their efforts, providing a sense of accomplishment and motivation for future contributions.

3. Improvement Feedback

- Description: Summarizes the participant’s contributions over the past 30 days and provides actionable feedback for further improvement.
- Examples:
    - "You have completed 10 technical support tasks in the past 30 days, meeting the criteria for the Tech Support NFT."
- Purpose: Serves as a guide for self-improvement, helping participants set their next goals and focus on growth.

Dashboard Use Cases

This UI is designed for the following purposes:

1. Visualizing Achievements:
    - Provides a clear overview of past activities and score trends, supporting self-assessment and motivation.
2. Reward Tracking:
    - Enables participants to review the NFTs and special items they have received, supporting goal setting for future rewards.
3. Improving Self-Awareness:
    - Provides feedback to help participants identify areas for improvement and set new objectives for continued growth.
4. Long-Term Engagement:

    - Encourages regular reflection on personal contributions, promoting sustained participation and self-development.

Operational Benefits of Customization

The customization of this participant dashboard offers several operational advantages:

- Clarity in UI Distribution:
    - Delivers only the necessary information to each stakeholder, reducing confusion and enhancing user experience.
- Enhanced Security:
    - Strictly controls access to score editing and personal data, ensuring secure user interactions.
- Participant Trust and Satisfaction:
    - Clearly presents the basis for score changes and reward issuance, fostering confidence in the system.
- Community-Wide Transparency:
    - Empowers both government and regional businesses to view relevant data, encouraging a sense of ownership and active participation.

This approach to role-based UI control and multi-environment optimization serves as a critical infrastructure element, supporting the scalability and credibility of local currency systems and municipal collaboration initiatives.

(7) Audit and Accountability – Log Reference Functionality

This system includes a Log Reference Functionality to ensure the transparency and reliability of reward distribution. It maintains a fully traceable record of all user actions, score changes, and reward decision processes on XTDB (a bitemporal database), allowing stakeholders to retrospectively track and verify complete event histories. This includes who performed what actions, when, how the scores changed, and what rewards were earned, all preserved in a comprehensive, time-sequenced format.

Key Features of the Log System

- Traceability of User Actions:
    - Every user action, score adjustment, and reward issuance is stored in a chronological log, providing a complete audit trail.

- Decision Logic Recording:
    - The system also logs the logic and reasoning behind reward distributions, including score thresholds and the application of time-limited campaigns.
    - This information is paired with natural language explanations generated by LLM (Large Language Models), allowing not just numerical transparency but also accountability for why certain rewards were granted.

Potential Use Cases

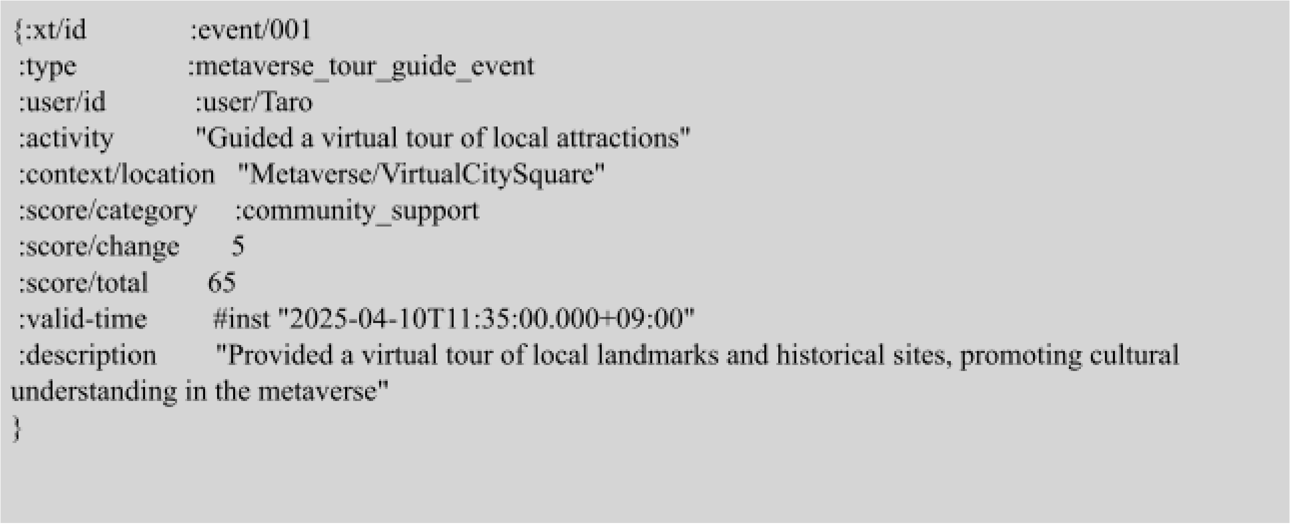
This functionality supports the following scenarios:

1. Audit Support for Municipalities and Currency Issuers:
    - Third-party auditors can review records for any period, individual, or rule to verify the fairness and legitimacy of reward distributions.
2. External Evaluation (Grants, Research Institutions):
    - Provides evidence-based reporting on the effectiveness of reward programs and their impact on residents.
3. User Self-Verification:
    - Users can review their own activity histories, including what actions were recognized and when, supporting confidence in the system and encouraging continuous improvement.
4. Rule Change Validation:
    - Allows for the comparison of distribution histories under different rule sets, providing insights for policy improvement and version control.

Example of Reference Data for an NFT Awarded User

For example, if a user who received an NFT asks "Which actions were the basis for this reward?", the system can reference the following XTDB event data:

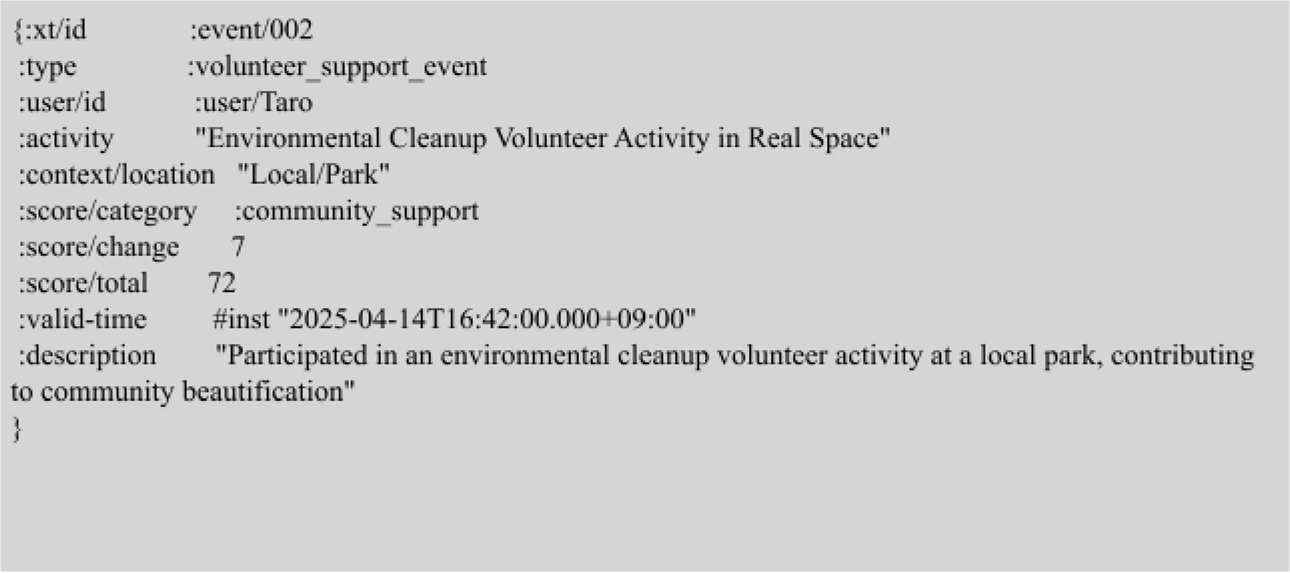
For XTDB, each event is stored with the following data format.



```
{:xt/id              :event/001
 :type               :metaverse_tour_guide_event
 :user/id            :user/Taro
 :activity           "Guided a virtual tour of local attractions"
 :context/location   "Metaverse/VirtualCitySquare"
 :score/category     :community_support
 :score/change       5
 :score/total        65
 :valid-time         #inst "2025-04-10T11:35:00.000+09:00"
 :description        "Provided a virtual tour of local landmarks and historical sites, promoting cultural
understanding in the metaverse"
}
```

Metaverse Tour Guide Event (April 10, 2025)
This data represents the following event:

Metaverse Tour Guide (April 10, 2025)
"On April 10, 2025, at 11:35 AM, the user "Taro" conducted a virtual tour in the metaverse space "VirtualCitySquare", guiding other participants through local attractions and historical landmarks. This activity was recorded as a metaverse_tour_guide_event, and based on the community support evaluation criteria, 5 points were added to the score. As a result, the cumulative score at that time reached 65 points."



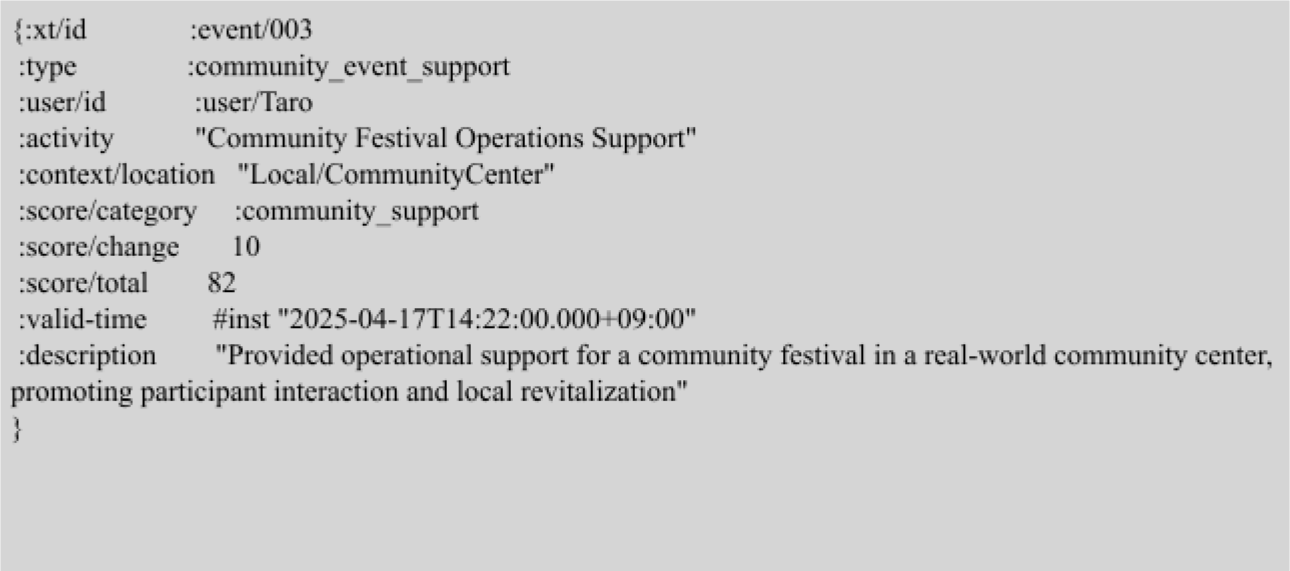
```
{:xt/id              :event/002
 :type               :volunteer_support_event
 :user/id            :user/Taro
 :activity           "Environmental Cleanup Volunteer Activity in Real Space"
 :context/location   "Local/Park"
 :score/category     :community_support
 :score/change       7
 :score/total        72
 :valid-time         #inst "2025-04-14T16:42:00.000+09:00"
 :description        "Participated in an environmental cleanup volunteer activity at a local park, contributing
to community beautification"
}
```

Real-World Volunteer Support (April 14, 2025)

This data represents the following event:

Real-World Volunteer Support (April 14, 2025)
"On April 14, 2025, at 4:42 PM, the user "Taro" participated in an environmental cleanup volunteer activity in a real-world local park, contributing to community beautification and

environmental preservation. This activity was recorded as a volunteer_support_event and, based on the evaluation criteria, 7 points were added to the community support score. As a result, Taro's cumulative score reached 72 points."



```
{:xt/id            :event/003
 :type             :community_event_support
 :user/id          :user/Taro
 :activity         "Community Festival Operations Support"
 :context/location   "Local/CommunityCenter"
 :score/category     :community_support
 :score/change      10
 :score/total       82
 :valid-time        #inst "2025-04-17T14:22:00.000+09:00"
 :description       "Provided operational support for a community festival in a real-world community center,
promoting participant interaction and local revitalization"
}
```

Community Event Support (April 17, 2025)

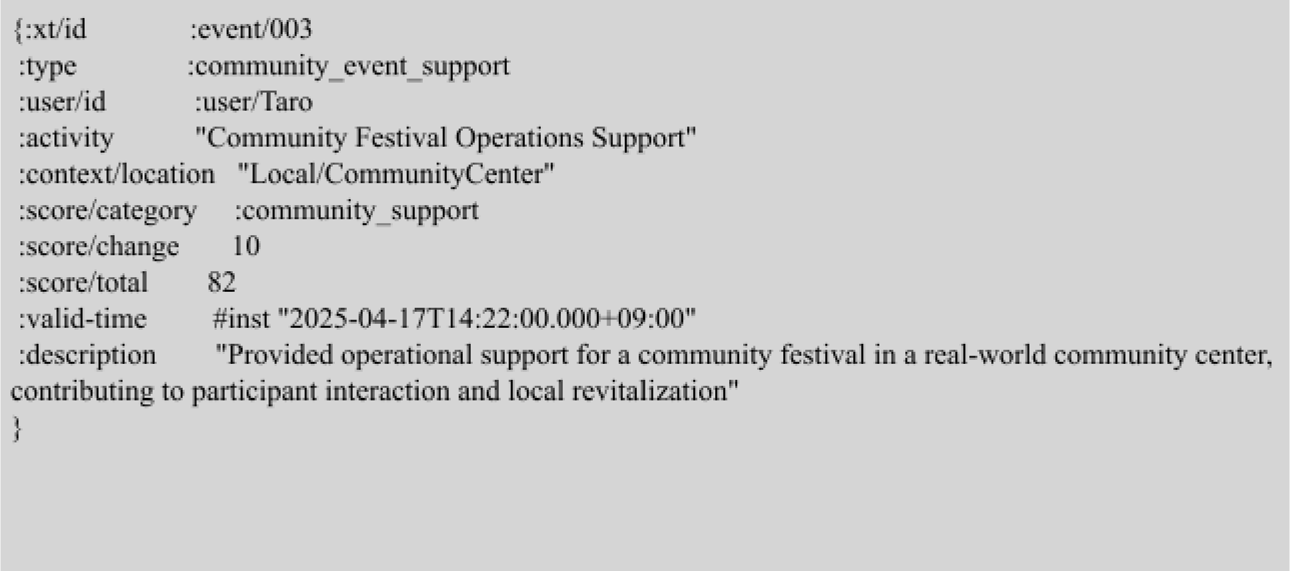
This data represents the following event:

LLM Support Event (April 17, 2025)
"On April 17, 2025, at 2:22 PM, the user "Taro" provided support in a metaverse environment, responding to a request for assistance on the effective use of Large Language Models (LLMs). Taro offered specific use case examples and relevant resources to the inquiring participant. This activity was recorded as an llm_support_event and, based on the evaluation criteria, 10 points were added to the technical support score. As a result, the cumulative score reached 82 points, surpassing the reward threshold of 80 points.

Additionally, at the time of this event, the user had completed 10 technical support activities within the last 30 days, fulfilling all the criteria required for NFT issuance, including:

- Cumulative technical support score of 80 points or more
- At least 10 support activities within the last 30 days

As such, this event served as the direct trigger for the subsequent NFT issuance process, becoming a critical historical event in the reward system."



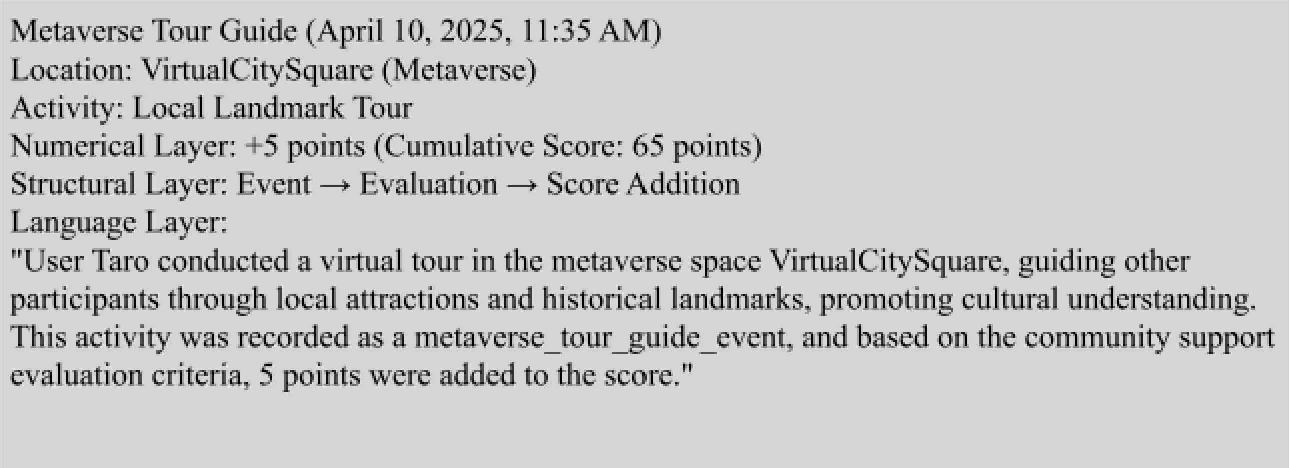
```
{:xt/id            :event/003
 :type             :community_event_support
 :user/id           :user/Taro
 :activity          "Community Festival Operations Support"
 :context/location   "Local/CommunityCenter"
 :score/category     :community_support
 :score/change      10
 :score/total       82
 :valid-time        #inst "2025-04-17T14:22:00.000+09:00"
 :description       "Provided operational support for a community festival in a real-world community center,
contributing to participant interaction and local revitalization"
}
```

NFT Issuance Event (April 17, 2025)

This data represents the following event:
"On April 17, 2025, at 2:23 PM, an NFT was officially issued to Taro as recognition for their consistent technical support contributions. This NFT issuance was automatically triggered by the previous :event/003 (LLM support event), where the required score and activity thresholds were satisfied."

NFT Issuance Event (April 17, 2025, 2:23 PM)

On April 17, 2025, at 2:23 PM, a "Community Support NFT" (Token ID: nft-community-support-0031) was officially issued to the user "Taro". This NFT was automatically granted in recognition of the user's ongoing contributions to community activities, including metaverse tour guides, real-world volunteer work, and community event support, after meeting the predefined issuance criteria.



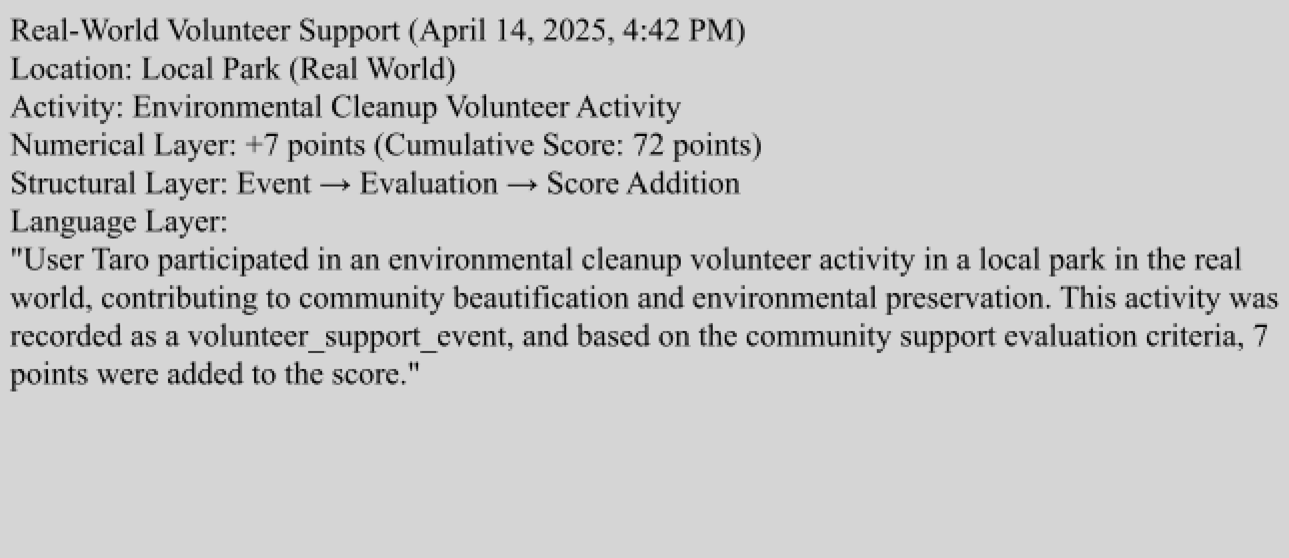
Metaverse Tour Guide (April 10, 2025, 11:35 AM)
Location: VirtualCitySquare (Metaverse)
Activity: Local Landmark Tour
Numerical Layer: +5 points (Cumulative Score: 65 points)
Structural Layer: Event → Evaluation → Score Addition
Language Layer:
"User Taro conducted a virtual tour in the metaverse space VirtualCitySquare, guiding other participants through local attractions and historical landmarks, promoting cultural understanding. This activity was recorded as a metaverse_tour_guide_event, and based on the community support evaluation criteria, 5 points were added to the score."

This NFT issuance was triggered by the preceding :event/003 (Community Event Support), where the score addition confirmed that all required issuance conditions were satisfied:

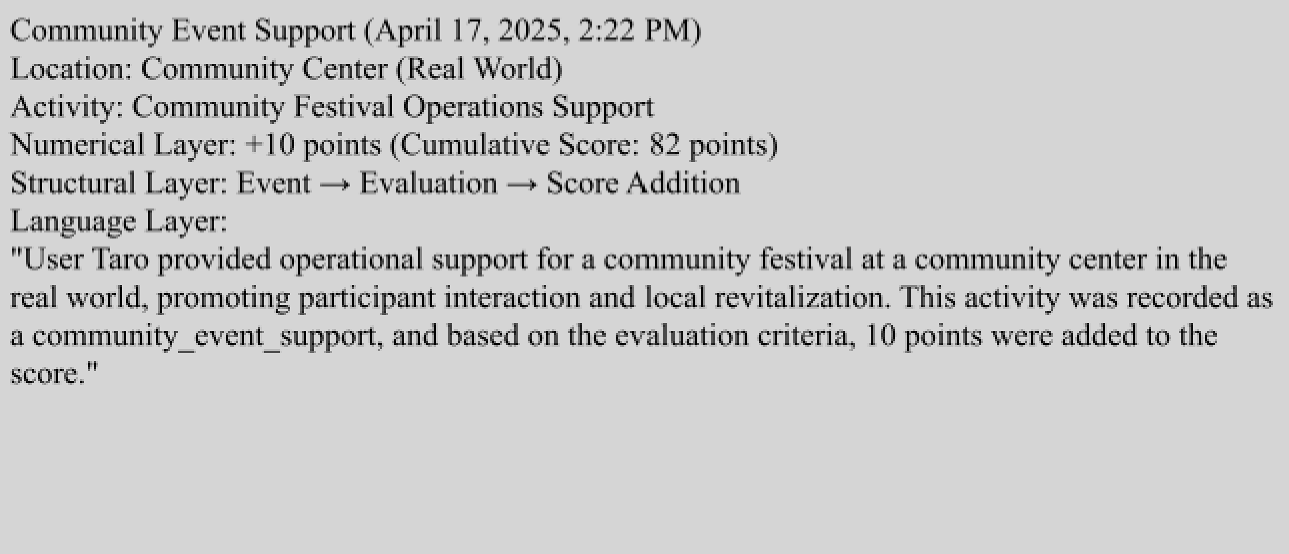
- Cumulative score of 80 points or more (Current score: 82 points)
- Participation in 10 or more community support events within the past 30 days
- Fulfillment of the threshold and timeframe requirements defined in the NFT issuance criteria

With all these conditions met, an nft_issuance_event was recorded in XTDB, and the token nft-community-support-0031 was assigned to the user.

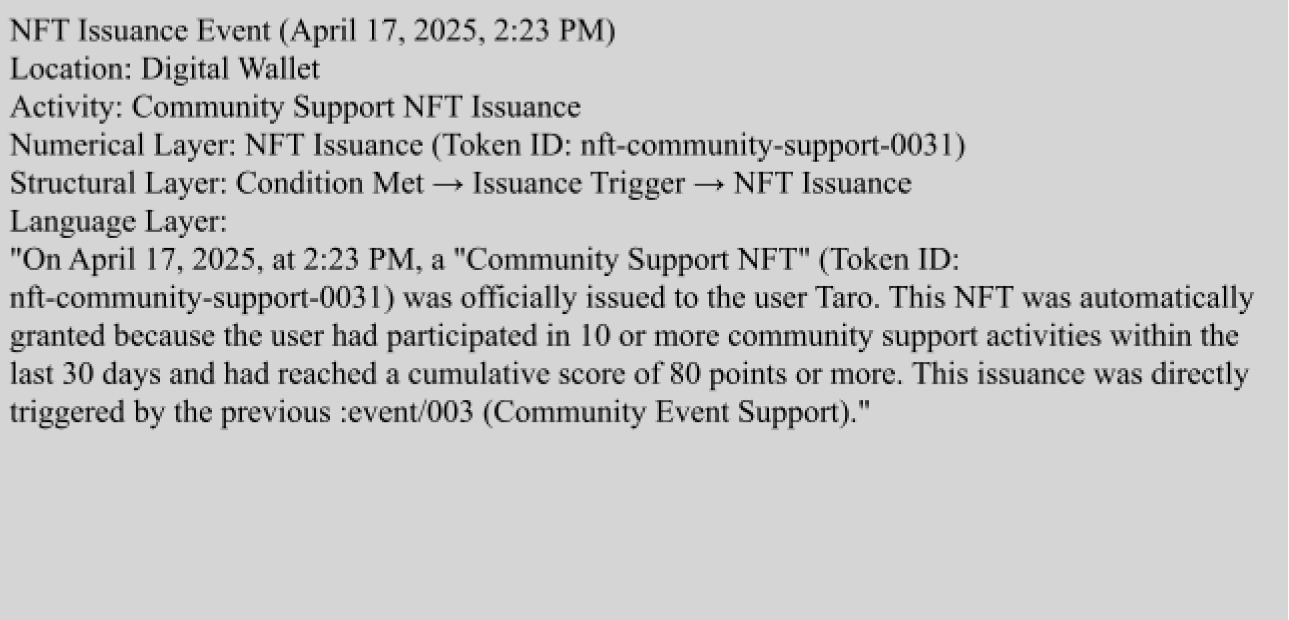
The results of visualizing these events with the View Agent are as follows.



Real-World Volunteer Support (April 14, 2025, 4:42 PM)
Location: Local Park (Real World)
Activity: Environmental Cleanup Volunteer Activity
Numerical Layer: +7 points (Cumulative Score: 72 points)
Structural Layer: Event → Evaluation → Score Addition
Language Layer:
"User Taro participated in an environmental cleanup volunteer activity in a local park in the real world, contributing to community beautification and environmental preservation. This activity was recorded as a volunteer_support_event, and based on the community support evaluation criteria, 7 points were added to the score."



Community Event Support (April 17, 2025, 2:22 PM)
Location: Community Center (Real World)
Activity: Community Festival Operations Support
Numerical Layer: +10 points (Cumulative Score: 82 points)
Structural Layer: Event → Evaluation → Score Addition
Language Layer:
"User Taro provided operational support for a community festival at a community center in the real world, promoting participant interaction and local revitalization. This activity was recorded as a community_event_support, and based on the evaluation criteria, 10 points were added to the score."



NFT Issuance Event (April 17, 2025, 2:23 PM)
Location: Digital Wallet
Activity: Community Support NFT Issuance
Numerical Layer: NFT Issuance (Token ID: nft-community-support-0031)
Structural Layer: Condition Met → Issuance Trigger → NFT Issuance
Language Layer:
"On April 17, 2025, at 2:23 PM, a "Community Support NFT" (Token ID: nft-community-support-0031) was officially issued to the user Taro. This NFT was automatically granted because the user had participated in 10 or more community support activities within the last 30 days and had reached a cumulative score of 80 points or more. This issuance was directly triggered by the previous :event/003 (Community Event Support)."

The View Agent functions not just as a visualization interface, but as a core visualization and analysis platform designed to optimize community engagement at a high level. Its essence lies in the advanced integration of the following three elements:

First, the LLM (Large Language Model) component provides contextual understanding of contribution behaviors, enabling the system to interpret and classify user actions and statements based on their social significance. This approach captures the human and collaborative value of contributions that cannot be fully represented by numerical scores alone.

Second, the GNN (Graph Neural Network) component dynamically analyzes network structures, identifying the structural roles that users play, such as central connectors, bridges, or supporters. This allows the system to assess and visualize influence, connectivity, and the quality of relationships within the community.

Third, the integration with XTDB provides robust support for score tracking, reward management, and behavioral history tracing. This enables precise quantification of policy effectiveness, temporal analysis of engagement, and transparent explanations of reward allocation, improving both user feedback and the accuracy of system design.

By combining these elements, the View Agent transcends the role of a mere visualization tool, emerging as an intelligent interface that promotes, contextualizes, and supports the sustainable formation of communities. It plays a critical role in creating the feedback loops necessary for sustained engagement and long-term community growth.

3. Future Prospects

The local currency system proposed in this white paper, built on the Cardano blockchain and Multi-Agent System (MAS), offers both technical and social scalability, with the potential for development in the following directions:

3.1 Governance Through Regional DAOs

- Introduction of Local DAOs (Decentralized Autonomous Organizations) that allow residents to participate in decision-making based on their token and NFT holdings.
- Proposal: Use decentralized voting to determine reward rules, policy priorities, and local event planning.
- Leverage Cardano smart contracts for secure voting processes and LLMs for automatic proposal generation and summarization, enabling democratic and efficient regional management.

3.2 Formation of Cross-Regional Currency Interchange Networks

- Development of cross-chain or common protocols to enable mutual exchange and use of tokens/NFTs between multiple municipalities or regions.
- Implementation of score conversion across different scoring systems using the Metric Analyzer Agent, allowing for the creation of a trusted, inter-regional credit score system where "Activities in City A are recognized in City B."

3.3 AI-Driven Initiative Optimization and Personalization

- Use LLMs and GNNs to provide personalized initiative recommendations for each resident (e.g., "This event is highly relevant to you").
- Analyze past behavior patterns, social network roles, and emotional tendencies to optimize individual interventions and predict their impact.

3.4 Application to Other Sectors (Education, Healthcare, Transportation)

- The scoring and blockchain-based reward systems can be extended to various fields:
    - Education: Scoring and validating achievements in exploratory learning, volunteer work, and peer reviews.
    - Healthcare and Welfare: Issuing gratitude tokens for caregiving and healthy behavior.
    - Transportation: Incentivizing public transportation use and congestion avoidance through token rewards.

3.5 Decentralized Identity (DID) and Personal Data Integration

- Integrate with Decentralized Identity (DID) technologies, allowing individuals to manage their own contribution histories, scores, and NFT holdings securely.
- Support seamless integration with wallets like MetaMask or mobile DID apps, balancing seamless user experience with privacy protection.

3.6 Strengthening Partnerships with Governments, Municipalities, and Enterprises

- Transition from proof-of-concept (PoC) trials by local governments to full-scale social implementation.
- Collaborate with businesses and shopping districts to develop commercial use cases for local token economies.

- Position this system as a model for "Decentralized Regional Activation Infrastructure" at the national level.

3.7 Open APIs and Ecosystem Development

- Provide APIs for external developers and local IT organizations, covering core components like the Metric Analyzer, Cardano, and View systems.
- Develop a Local Economy SDK to enable rapid implementation of local applications and event integrations.
- Promote the creation of an open ecosystem for regional economies through collaboration across multiple projects.

In this way, the proposed system has the potential to evolve beyond a simple local currency management platform into a Distributed Civic Infrastructure that supports both physical and virtual community building. The goal is to achieve a seamless blend of real-world, virtual, and institutional systems, ultimately fostering a sustainable, locally engaged economy.

4. Conclusion

This white paper has proposed a decentralized operational framework for a new local currency system that integrates the Cardano blockchain and Multi-Agent Systems (MAS). By leveraging real-time scoring of multi-dimensional contributions—such as actions, knowledge, relationships, and enthusiasm—this system enables the automatic design and issuance of rewards like NFTs and tokens, creating a sustainable and highly transparent regional economy.

Core Architecture Elements

At the heart of this system is an integrated architecture combining the following key elements:

- Autonomous Contribution Evaluation and Reward Distribution via MAS:
    - Real-time, agent-driven scoring of non-monetary and informal contributions, capturing the true value of each participant.
- Immutable Asset Recording and Contract Execution on the Cardano Blockchain:
    - Secure, tamper-proof records of transactions and agreements, providing a reliable foundation for community economies.
- Semantic Understanding and Context Generation via LLMs, and Structural Analysis via GNNs:

    - Combines the interpretive power of large language models (LLMs) with the structural insight of graph neural networks (GNNs) to assess social interactions.

- Time-Series Tracking and Causal Analysis via XTDB:
    - Maintains a comprehensive record of contribution histories, enabling deep analysis of cause-effect relationships over time.
- Visualization, Explanation, and KPI Support via View Agents:
    - Provides intuitive, real-time insights into community contributions, enhancing transparency and decision-making.

Towards a New Model of Local Communities

This integrated approach captures previously invisible, non-monetary contributions as valuable assets, enabling each resident to be recognized fairly and meaningfully for their participation. It fosters a new model of local society where individuals experience the joy of being valued as contributors.

Moreover, this proposal extends beyond the digitalization of local currency. It envisions a next-generation civic infrastructure, incorporating:

- Decentralized Governance via DAOs
- Cross-Regional Currency Networks
- Applications in Education, Healthcare, and Public Transportation

Moving forward, the focus will be on optimizing operational models and verifying system reliability through real-world pilots, aiming for a scalable, decentralized economic infrastructure driven by local communities.

5. References

- M.Watanabe:https://deus-ex-machina-ism.com/
- Greco, Thomas H. (2009). The End of Money and the Future of Civilization. Chelsea Green Publishing.
  — A comprehensive guide to the economic and social significance of local currencies.
- Swan, Melanie (2015). Blockchain: Blueprint for a New Economy. O'Reilly Media.
  — A pioneering discussion on the potential of blockchain as a social infrastructure.
- Lietaer, Bernard, & Dunne, Jacqui (2013). Rethinking Money: How New Currencies Turn Scarcity into Prosperity.Berrett-Koehler Publishers.
  — Examines the role of complementary currencies in addressing economic instability.

- Dini, Paolo, & Kioupkiolis, Alexandros (2019). The Political Economy of Blockchain. University of Westminster Press.
  — Analyzes the socio-economic impacts of blockchain in local and global contexts.

- Wooldridge, Michael (2009). An Introduction to MultiAgent Systems (2nd Edition). Wiley.
  — A foundational text on the design theory, agent cooperation, and distributed decision-making in MAS.

- Shoham, Yoav, & Leyton-Brown, Kevin (2008). Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations. Cambridge University Press.
  — A mathematical foundation for strategic interactions among agents, including game theory and auction design.

- Weiss, Gerhard (2013). Multiagent Systems (2nd Edition). MIT Press.
  — A comprehensive overview of MAS design, including communication protocols and cooperative strategies.

- Jennings, Nicholas R., & Wooldridge, Michael (1998). Agent Technology: Foundations, Applications, and Markets.Springer.
  — An early exploration of the practical applications of agent-based systems.

- Narayanan, Arvind et al. (2016). Bitcoin and Cryptocurrency Technologies. Princeton University Press.
  — A technical foundation for understanding distributed consensus and the underlying technologies of cryptocurrencies.

- Plutus Pioneer Program by IOHK/Cardano
  — Educational materials for developing smart contracts on the Cardano blockchain. (Available at: Plutus Pioneer Program)

- Mougayar, William (2016). The Business Blockchain: Promise, Practice, and the Application of the Next Internet Technology. Wiley.
  — A practical guide to implementing blockchain in business contexts.

- Tapscott, Don, & Tapscott, Alex (2016). Blockchain Revolution: How the Technology Behind Bitcoin and Other Cryptocurrencies is Changing the World. Penguin.
  — A comprehensive look at the transformative potential of blockchain technology.

- Mitchell, Melanie (2019). Artificial Intelligence: A Guide for Thinking Humans. Farrar, Straus and Giroux.
  — A critical examination of the impacts of AI technologies like LLMs on human society.

- Kipf, Thomas & Welling, Max (2016). Semi-Supervised Classification with Graph Convolutional Networks.arXiv:1609.02907.
  — The foundational paper on Graph Convolutional Networks (GCNs), essential for social graph analysis.

- Bommasani, R. et al. (2021). On the Opportunities and Risks of Foundation Models. Stanford CRFM.
  — A comprehensive review of the societal implications of foundational models like GPT and BERT.

- Goodfellow, Ian, Bengio, Yoshua, & Courville, Aaron (2016). Deep Learning. MIT Press.
  — A foundational text covering the core concepts of deep learning, including neural networks and LLM architectures.

- Russell, Stuart, & Norvig, Peter (2021). Artificial Intelligence: A Modern Approach (4th Edition). Pearson.
  — A definitive guide to the field of artificial intelligence, including planning, perception, and machine learning.

- Stonebraker, Michael (2018). The 12 Rules of OLAP. VLDB.
  — Principles for designing effective data visualization and analytical dashboards.

- XTDB Official Documentation (XTDB Docs)
  — Comprehensive resources on bitemporal database architecture, data reconstruction, and change history management.

- Yau, Nathan (2013). Data Points: Visualization That Means Something. Wiley.
  — Practical guidance on integrating visualization with narrative for impactful data presentation.

- Inmon, Bill (2005). Building the Data Warehouse (4th Edition). Wiley.
  — A classic text on data warehousing and data architecture design.

- Kimball, Ralph, & Ross, Margy (2013). The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling (3rd Edition). Wiley.
  — A practical guide to building dimensional data models for complex analytical systems.