2. Certificate IPDC transmission support tool [2]

2.1 Function Description

This system is a web-based tool that leverages the Cardano blockchain to issue and manage verifiable certificates for disaster relief volunteers.

Each **certificate** includes detailed information such as:
・Volunteer skills
・Volunteer Roles
・Volunteer Authorized regions

By using this system, disaster response organizations can conduct reliable and transparent credential verification, ensuring trust and efficiency in emergency operations.

Key features of the system include:

2.1.1 IPDC Transmission Data Construction Function [2-a]

2.1.1.1 Periodic Block Information Retrieval

- The tool retrieves the latest block information from the Cardano blockchain every 15 seconds. It uses the Blockfrost API to fetch block data, including:
  ・Block height
  ・Block hash
  ・Previous block information
- A file lock mechanism ensures that if a specified file already exists, the data retrieval process is skipped. This prevents conflicts when running concurrent processes.

2.1.1.2 Block Integrity Check and Anomaly Detection

- The system verifies whether the **latest block information** matches the **previously retrieved block** and ensures the **block hash is correct**. If an **anomaly is detected**, a **reconciliation process** is triggered, utilizing historical block data to verify and maintain **data consistency**.
- In case of **discrepancies**, an **anomaly message** is logged to ensure **data reliability and trustworthiness**.

2.1.1.3 Generation and Storage of IPDC Hash Information

- Each block is assigned IPDC-compliant hash information, allowing the next block to reference the previous IPDC hash in accordance with the IPDC format. The sha3_256 algorithm is used to generate the IPDC hash for each block.
- The new IPDC hash is calculated by combining:
  - The previous IPDC hash
  - The current block's transaction information
- The computed IPDC hash information is stored in a file and can be used as proof data for each block.

2.1.1.4 Merkle Root Calculation for Transactions

- Retrieves the list of transaction IDs from the latest block and generates a Merkle tree to verify the integrity of block data.
- The Merkle root hash is added to each block and is also used in IPDC hash calculations.
- If no transactions exist in a block, an empty Merkle root hash is set to ensure data integrity.

2.1.1.5 Certificate File Creation

- Block information and the computed IPDC hash are stored in a JSON file, allowing for later reference and verification.
- A certificate file is generated for each block height, ensuring data traceability.
- The latest file name is appended to trigger.txt, which serves as a trigger file for integration with other systems.

2.1.1.6 Log Management and Error Handling

- Uses the **Log4js library** to log key steps in the process, including:
  - **Tool startup**
  - **Data retrieval**
  - **Anomaly detection**
  - **IPDC hash generation**
  - **File creation**
- These logs help with **troubleshooting and debugging**.
- When an **error occurs**, a **detailed error message** is logged to facilitate **problem detection and resolution**.

2.1.1.7 Automatic Deletion of Old Files

- **Automatically deletes old files** that have exceeded a **specified retention period**, ensuring **efficient storage management**.
- By **default**, file deletion is executed **once per day**, keeping the **storage directory clean and optimized**.

2.1.1.8 Use Cases

- **Real-time Blockchain Data Monitoring**
  Periodically retrieves the latest Cardano blockchain data and generates certificates in IPDC hash format, ensuring that each block's data is linked. Instant anomaly detection helps maintain blockchain data integrity by identifying inconsistencies in real time.
- **Data Proof via IPDC Integration**
  Manages blockchain data in IPDC format, ensuring trustworthiness when referenced by other systems. The generated certificate files provide tamper-proof records, enabling decentralized data verification.

Thus, the Certificate IPDC Transmission Support Tool is designed to periodically retrieve and verify data on the Cardano blockchain while ensuring high reliability and data integrity.

Through IPDC-format hash management, file storage, and anomaly detection, this tool enables secure and consistent block data management.

2.2 System Architecture

The Certificate IPDC Transmission Support Tool is a system that periodically retrieves the latest block information from the Cardano blockchain and assigns an IPDC-format hash to each block to ensure data linkage consistency.

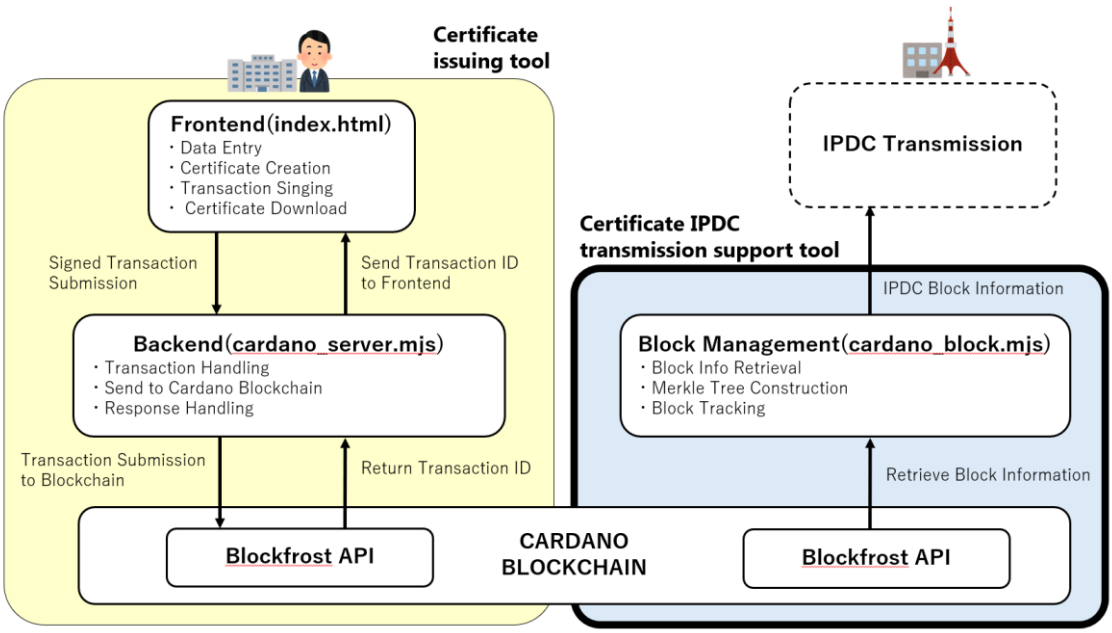
This tool:

・ Manages and stores retrieved block information as IPDC hashes.

・ Generates certificate files in a format that allows integration with other systems.

2.2.1 System Architecture Diagram

This system consists of a blockchain information update and management module (cardano_block.mjs) and operates in coordination with the Cardano blockchain and the Blockfrost API. It periodically retrieves IPDC block information from the Cardano blockchain and provides the data to the broadcasting station for IPDC transmission.



2.2.2 System Components

2.2.2.1 Block Information Retrieval (Blockfrost API)

- Utilizes the Blockfrost API to fetch the latest Cardano blockchain block information every 15 seconds.
- Each block's data includes:
  - ・Block hash
  - ・Previous block information
  - ・Transaction list within the block
- This information ensures data integrity and consistency across the system.
- Implements a file lock mechanism to pause data retrieval if a specified file exists, preventing conflicts with other processes.

2.2.2.2 IPDC Hash Generation

- Based on the retrieved block data, an IPDC-format hash is generated for each block.
- The sha3_256** algorithm** is used to create the hash by combining:
  - ・The previous block's IPDC hash
  - ・The current block's data
- The generated IPDC hash is referenced by the next block, ensuring data continuity and reliability.

2.2.2.3 File Storage and Trigger File Update

- A certificate file containing the calculated IPDC hash and block information is generated in JSON format and stored for each block. This ensures that each block's integrity is recorded as verified data.
- The certificate file name is appended to the trigger.txt file, making it easier for external systems to detect newly generated certificate files.
- By updating the trigger file, the system supports efficient data retrieval for external integrations.

2.2.2.4 Log Management

- Utilizes the Log4js library to log information on data retrieval, IPDC hash generation, certificate file creation, anomaly detection, and error handling.
- Each log entry contains processing execution status and detailed error information, which helps monitor the tool's operation and facilitates troubleshooting.

2.2.2.5 Automatic Deletion of Old Files

- Certificate files that exceed a specified retention period will be automatically deleted to efficiently manage the system's storage.
- The deletion process is executed once per day, ensuring proper organization of files that do not require long-term storage.

2.2.2.6 Data Flow

- **Retrieving Block Information**
  The latest block information on the Cardano blockchain is retrieved via the Blockfrost API, ensuring consistency with the previously recorded block information.
- **Generating IPDC Hash**
  Based on the IPDC hash of the previous block, a new IPDC hash is calculated using the retrieved block information.
- **Generating and Storing Certificate Files**
  The block information, including the IPDC hash, is stored in a JSON-format certificate file, and the trigger file is updated.
- **Logging and Error Handling**
  Data retrieval, processing results, and error occurrences are recorded in logs.
- **Deleting Old Files**
  Old files that exceed the retention period are regularly deleted to manage storage efficiently.

With this structure, the Certificate IPDC Transmission Support Tool consistently processes everything from retrieving block information to generating the IPDC hash and storing data, ensuring overall data integrity within the system.

2.3 Module Design

This tool consistently performs tasks such as retrieving block information, generating IPDC hashes, saving certificate files, and managing logs within a single module. The roles and processing details of each function are described below.

2.3.1 Module Processing Details

The frontend is responsible for retrieving block information, generating IPDC hashes, creating and saving certificate files, managing logs, and implementing an automatic file deletion feature. The main functions and their roles are as follows:

2.3.1.1 Retrieving Block Information

- **Processing Details**

  The system accesses the Blockfrost API at 15-second intervals to retrieve the latest block information from the Cardano blockchain.

- **Retrieved Data**
  - Latest block hash,
  - Previous block information,
  - Transaction list within the block.

- **File Lock Feature**

  If a specified file exists, the data retrieval process is skipped to prevent conflicts with other processes.

- **Error Handling**

  In case of API request failures, error messages are logged, and the retrieval process is retried in the next cycle.

2.3.1.2 Generating IPDC Hash

- **Processing Details**

  A new IPDC hash is generated by combining the retrieved latest block information with the previously generated IPDC hash.

- **Generation Method**

  The sha3_256 algorithm is used to concatenate the previous IPDC hash with the current block data and compute the new IPDC hash.

  The generated IPDC hash is also used as a reference hash for the next block, ensuring data integrity and continuity.

- **Error Handling**

  If hash generation fails, an error log is recorded, and the process is retried in the next retrieval cycle.

2.3.1.3 Generating and Saving Certificate Files

- **Processing Details**

  The generated IPDC hash and block information are saved as a certificate file in JSON format.

- **File Structure**

  A JSON file is created for each block, containing block height, block hash, transaction list, and IPDC hash.

  The certificate file name is appended to **"trigger.txt"**, enabling external systems to detect file generation.

- **Error Handling**

  If file generation or saving fails, an error log is recorded, and the process is retried in the next cycle.

2.3.1.4 Log Management

- **Processing Details**

  Execution results of each process and error details in case of abnormalities are recorded in logs.

- **Log Content**

  During normal operation, logs record the start and end of processes and successful data retrieval statuses.

  In case of abnormalities, logs store detailed error information, including error messages, location of occurrence, and timestamps.

- **Log Library**

  The **Log4js** library is used to efficiently manage operational information.

2.3.1.5 Automatic Deletion of Old Files

- **Processing Details**

  The system periodically checks the storage directory and deletes certificate files older than a specified number of days.

- **Execution Timing**

  This process runs once per day to automatically delete unnecessary files and maintain efficient storage usage.

- **Error Handling**

  If deletion fails, an error log is recorded, and the process is retried in the next execution cycle.

2.3.1.6 Overall Module Workflow

- **Processing starts every 15 seconds**

  Retrieves block information and generates an IPDC hash.

- **Certificate file generation**

  Creates a certificate file using the retrieved block information and IPDC hash, then updates the trigger file.

- **Logging**

  Records the status of each process and error handling results.

- **Daily automatic deletion**

  Deletes old files exceeding the specified retention period to manage system storage.

By following this structure, the IPDC Certificate Transmission Support Tool ensures seamless processing from block information retrieval to IPDC hash generation and data storage, maintaining overall data integrity within the system.