



(19) 대한민국특허청(KR)  
(12) 등록특허공보(B1)

(45) 공고일자 2019년09월27일  
(11) 등록번호 10-2026225  
(24) 등록일자 2019년09월23일

(51) 국제특허분류(Int. Cl.)  
H04L 9/06 (2006.01) H04L 29/08 (2006.01)  
H04L 9/08 (2006.01)  
(52) CPC특허분류  
H04L 9/0637 (2013.01)  
H04L 67/02 (2013.01)  
(21) 출원번호 10-2017-0009285  
(22) 출원일자 2017년01월19일  
심사청구일자 2017년01월19일  
(65) 공개번호 10-2018-0085570  
(43) 공개일자 2018년07월27일  
(56) 선행기술조사문헌  
US20170005804 A1\*  
KR101637854 B1  
KR101628005 B1  
KR101590076 B1  
\*는 심사관에 의하여 인용된 문헌

(73) 특허권자  
주식회사 케이티  
경기도 성남시 분당구 불정로 90(정자동)  
(72) 발명자  
용창욱  
경기도 용인시 기흥구 연원로42번길 2, 120동  
1003호 (마북동, 연원마을벽산아파트)  
김중철  
경기도 성남시 수정구 위례동로 61, 5618동 1402  
호 (창곡동, 위례 자연엔 래미안이편한세상)  
유정석  
서울특별시 강남구 광평로51길 27, 405동 1308호  
(수서동, 삼익아파트)  
(74) 대리인  
특허법인필앤은지

전체 청구항 수 : 총 20 항

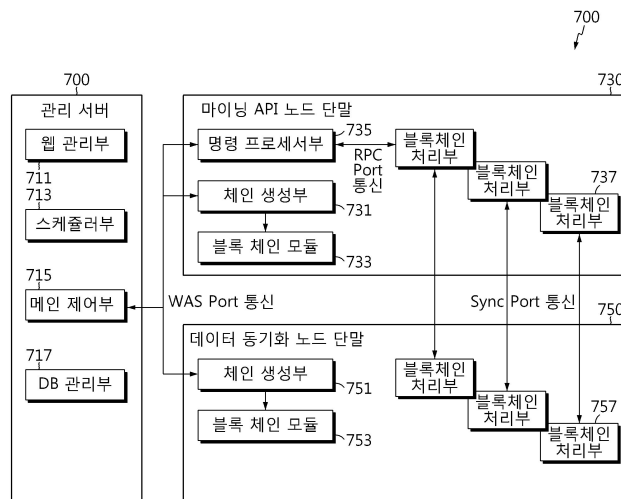
심사관 : 양종필

(54) 발명의 명칭 블록 체인을 이용하여 데이터를 관리하는 장치 및 방법

(57) 요약

블록 체인에 저장되는 데이터를 관리하는 장치 및 방법이 개시된다. 일 측면에 따른, 복수의 노드 단말들이 데이터를 동기화하여 저장하기 위해 형성한 블록 체인 및 상기 블록 체인의 데이터의 저장, 조회 및 삭제를 관리하는 장치는, 노드 단말로 블록 체인을 관리하는 체인 명령 및 데이터를 관리하는 데이터 명령을 전송하여 데이터의 관리를 제어하는 메인 제어부; 및 체인 명령에 의해 생성된 체인 정보를 체인 테이블에 저장하고, 데이터 명령에 의해 저장된 데이터 및 트랜잭션의 매핑 정보를 매핑 테이블에 저장하는 DB 관리부를 포함한다.

대표도 - 도7



(52) CPC특허분류  
*H04L 9/0894* (2013.01)

---

## 명세서

### 청구범위

#### 청구항 1

복수의 노드 단말들이 데이터를 동기화하여 저장하기 위한 블록 체인의 생성 및 상기 블록 체인의 데이터의 저장, 조회 및 삭제를 관리하는 장치에 있어서,

소정의 기간 단위마다 서로 다른 블록 체인을 생성하기 위한 체인 명령을 상기 복수의 노드 단말 중 적어도 하나의 노드 단말로 전송하며 각 블록 체인마다 아이디(ID)를 생성하고, 특정 기간에 발생한 데이터의 아이디를 생성한 후 해당 특정 기간에 대응하는 블록 체인에 해당 데이터를 저장하기 위한 데이터 명령을 상기 복수의 노드 단말 중 적어도 하나의 노드 단말로 전송하고 이에 대한 응답으로 트랜잭션 아이디를 수신하는 메인 제어부; 및

상기 각 블록 체인마다의 아이디를 체인 테이블에 저장하여 관리하고, 상기 데이터의 아이디, 상기 트랜잭션 아이디 및 저장 시간의 매핑 정보를 매핑 테이블에 저장하여 관리하는 DB 관리부를 포함하고,

상기 메인 제어부는, 상기 각 블록 체인의 아이디 생성시 각 블록 체인에 대응하는 기간의 정보를 이용하여 생성하는 장치.

#### 청구항 2

제 1항에 있어서,

상기 블록 체인은,

권한이 정의된 사용자에게만 데이터 접근이 허락되는 Private 모드를 제공하는 제 1기능;

블록 체인의 데이터 처리에 수수료가 없는 제 2기능;

블록 체인에 요구되는 노드 단말에 대한 마이닝 노드의 작업 증명 과정에서 노드의 검증 처리를 제거하고 데이터의 유효성 검증 처리를 수행하는 제 3기능;

상기 작업 증명 과정에 의해 단축된 블록의 생성 시간을 지연시키기 위해 딜레이를 주어 블록 생성 시간을 지연시키는 제 4기능; 및

저장 대상의 트랜잭션이 발생된 경우에만 블록을 생성하는 제 5기능

중에서 적어도 하나 이상의 기능이 설정되어 각각의 노드 단말에서 실행되는 것을 특징으로 하는 장치.

#### 청구항 3

제 1항에 있어서,

웹 페이지를 통해 사용자 인터페이스를 제공하여 상기 블록 체인 및 상기 데이터의 관리 정보를 입력받고, 입력된 관리 정보의 처리 결과를 제공하는 웹 관리부를 더 포함하는 것을 특징으로 하는 장치.

#### 청구항 4

제 1항에 있어서,

상기 블록 체인 및 상기 데이터의 관리를 위한 스케줄링 정보를 등록받고, 등록된 스케줄링 정보에 대응되는 프로세스의 실행을 상기 메인 제어부로 요청하는 스케줄러부를 더 포함하고,

상기 메인 제어부는 상기 요청에 의해 대응되는 명령을 전송하는 것을 특징으로 하는 장치.

#### 청구항 5

제 1항에 있어서,

상기 노드 단말은,

블록을 생성해서 블록 체인에 연결하는 마이닝 처리 및 블록 체인의 데이터 정보를 API(Application program interface)를 통해 제공하는 마이닝 API 노드 단말; 및

마이닝 API 노드 단말에 의해 블록 체인의 데이터 관리가 동기화되는 데이터 동기화 노드 단말로 구성되는 것을 특징으로 하는 장치.

#### 청구항 6

제 5항에 있어서,

상기 마이닝 API 노드 단말은,

상기 체인 명령을 전송받고, 전송받은 체인 명령에 따른 블록 체인의 생성, 삭제 및 실행을 처리하고, 처리 결과를 응답하는 체인 생성부

상기 데이터 명령을 전송받고, 전송받은 데이터 명령을 처리하고, 처리 결과를 응답하는 명령 프로세서부; 및

상기 처리에 따른 데이터 명령을 블록 체인에 대해 실행하는 블록 체인 처리부

를 포함하는 것을 특징으로 하는 장치.

#### 청구항 7

제 6항에 있어서,

상기 데이터 동기화 노드 단말은,

상기 체인 명령을 전송받고, 전송받은 체인 명령에 따른 블록 체인의 생성, 삭제 및 실행을 처리하고, 처리 결과를 응답하는 체인 생성부; 및

상기 마이닝 API 노드 단말의 블록 체인 처리부와 포트 연결되고, 연결된 포트를 통해 전송된 데이터 명령을 동기화하여 실행하는 블록 체인 처리부

를 포함하는 것을 특징으로 하는 장치.

#### 청구항 8

제 1항에 있어서,

상기 메인 제어부는,

상기 블록 체인의 아이디에 기초하여, 상기 블록 체인과 통신하기 위한 포트 식별정보와, 상기 복수의 노드 단말 간의 동일 블록 체인의 동기화를 위한 포트 식별정보와, 상기 데이터가 저장될 디렉토리 경로를 생성하는 것을 특징으로 하는 장치.

#### 청구항 9

제 8항에 있어서,

상기 메인 제어부는,

저장 대상의 데이터의 파일 정보에 포함된 시간 정보를 이용하여 상기 체인 테이블로부터 블록 체인의 아이디를 조회하고, 조회된 블록 체인의 아이디에 기초하여 상기 데이터 명령을 상기 적어도 하나의 노드 단말로 전송하여 데이터 저장을 요청하는 것을 특징으로 하는 장치.

#### 청구항 10

삭제

#### 청구항 11

제 9항에 있어서,

상기 메인 제어부는,

조회 대상의 데이터의 파일 정보를 입력받고, 입력된 파일 정보에 대응되는 매핑 정보를 상기 매핑 테이블로부터 조회하고, 조회된 매핑 정보에 따른 데이터 조회를 위한 데이터 명령을 상기 적어도 하나의 노드 단말로 전송하여 데이터 조회를 요청하고, 조회 결과를 수신하는 것을 특징으로 하는 장치.

**청구항 12**

복수의 노드 단말들이 데이터를 동기화하여 저장하기 위한 블록 체인의 생성 및 상기 블록 체인의 데이터를 관리하는 장치가 데이터를 저장하는 방법에 있어서,

소정의 기간 단위마다 서로 다른 블록 체인을 생성하기 위한 체인 명령을 상기 복수의 노드 단말 중 적어도 하나의 노드 단말로 전송하며 각 블록 체인마다 아이디(ID)를 생성하되, 각 블록 체인의 아이디를 각 블록 체인에 대응하는 기간의 정보를 이용하여 생성하는 단계;

상기 각 블록 체인마다의 아이디를 체인 테이블에 저장하는 단계;

특정 기간에 발생한 데이터의 아이디를 생성한 후 해당 특정 기간에 대응하는 블록 체인에 해당 데이터를 저장하기 위한 데이터 명령을 상기 복수의 노드 단말 중 적어도 하나의 노드 단말로 전송하고 이에 대한 응답으로 트랜잭션 아이디를 수신하는 단계; 및

상기 데이터의 아이디, 상기 트랜잭션 아이디 및 저장 시간의 매핑 정보를 매핑 테이블에 저장하는 단계를 포함하는 방법.

**청구항 13**

제 12항에 있어서,

상기 블록 체인은,

권한이 정의된 사용자에게만 데이터 접근이 허락되는 Private 모드를 제공하는 제 1기능;

블록 체인의 데이터 처리에 수수료가 없는 제 2기능;

블록 체인에 요구되는 노드 단말에 대한 마이닝 노드의 작업 증명 과정에서 노드의 검증 처리를 제거하고 데이터의 유효성 검증 처리를 수행하는 제 3기능;

상기 작업 증명 과정에 의해 단축된 블록의 생성 시간을 지연시키기 위해 딜레이를 주어 블록 생성 시간을 지연시키는 제 4기능; 및

저장 대상의 트랜잭션이 발생된 경우에만 블록을 생성하는 제 5기능

중에서 적어도 하나 이상의 기능이 설정되어 각각의 노드 단말에서 실행되는 것을 특징으로 하는 방법.

**청구항 14**

제 12항에 있어서,

상기 생성하는 단계는,

상기 데이터의 파일명을 이용하여 고유한 블록 체인의 아이디를 생성하는 것을 특징으로 하는 방법.

**청구항 15**

제 12항에 있어서,

상기 생성하는 단계는,

상기 체인 테이블로부터 상기 체인 아이디에 대응하는 정보를 조회하는 단계를 포함하는 것을 특징으로 하는 방법.

**청구항 16**

제 15항에 있어서,

상기 생성하는 단계는,

상기 조회하는 단계에서 조회가 실패될 경우, 상기 체인 아이디를 갖는 블록 체인의 생성을 위한 상기 체인 명령을 상기 적어도 하나의 노드 단말로 전송하는 것을 특징으로 하는 방법.

**청구항 17**

제 12항에 있어서,

상기 트랜잭션 아이디를 수신하는 단계는,

상기 데이터의 파일 정보에 포함된 시간 정보를 이용하여 상기 체인 테이블로부터 블록 체인의 아이디를 조회하고, 조회된 블록 체인의 아이디에 기초하여 상기 데이터 명령을 상기 적어도 하나의 노드 단말로 전송하여 데이터 저장을 요청하는 단계를 포함하는 것을 특징으로 하는 방법.

**청구항 18**

제 12항에 있어서,

상기 복수의 노드 단말은,

상기 장치로부터 체인 명령 및 데이터 명령을 수신하여 처리하는 마이닝 API 단말 및 상기 장치로부터 체인 명령을 수신하여 처리하고, 상기 마이닝 API 단말의 데이터 명령이 동기화 처리되는 데이터 동기화 단말로 구성되고,

상기 트랜잭션 아이디를 수신하는 단계는,

상기 마이닝 API 단말이 상기 장치로부터 상기 데이터 명령을 전송받고, 전송받은 데이터 명령의 실행에 의해 블록 체인에 상기 데이터를 저장하는 단계;

상기 데이터 동기화 단말이 상기 마이닝 API 단말에 연결된 포트를 통해 전송된 데이터 명령의 동기화 실행에 의해 블록 체인에 상기 데이터를 저장하는 단계; 및

상기 마이닝 API 단말이 실행된 데이터 명령에 의한 상기 저장 결과로서 상기 트랜잭션 아이디를 상기 장치로 전송하는 단계

를 포함하는 것을 특징으로 하는 방법.

**청구항 19**

제 12항에 있어서,

상기 생성하는 단계는,

상기 블록 체인의 아이디에 기초하여, 상기 블록 체인과 통신하기 위한 포트 식별정보와, 상기 복수의 노드 단말 간의 동일 블록 체인의 동기화를 위한 포트 식별정보와, 상기 데이터가 저장될 디렉토리 경로를 생성하는 단계를 포함하는 것을 특징으로 하는 방법.

**청구항 20**

제 12항에 있어서,

상기 저장하는 단계 이후에,

스케줄러에 의해 설정된 조건이 만족되면, 조건이 만족된 블록 체인 및 데이터를 삭제하는 단계를 더 포함하는 것을 특징으로 하는 방법.

**청구항 21**

제 12항에 있어서,

조회 대상의 데이터의 파일 정보를 입력받고, 입력된 파일 정보에 대응되는 매핑 정보를 상기 매핑 테이블로부터 조회하고, 조회된 매핑 정보에 따른 데이터 조회를 위한 데이터 명령을 상기 노드 단말로 전송하여 데이터

조회를 요청하고, 조회 결과를 수신하는 단계를 더 포함하는 것을 특징으로 하는 방법.

**발명의 설명**

**기술 분야**

[0001] 본 발명은 데이터의 관리 기술로서, 보다 구체적으로, 블록 체인을 이용하여 데이터의 저장, 조회 및 삭제 등을 관리하는 장치 및 방법에 관한 것이다.

**배경 기술**

[0002] 종래의 파일 시스템에서는 디스크에 블록 사이즈(예 : 8KB)를 지정하여 포맷한 상태에서 사용한다. 참고로, 유닉스 파일 시스템에서는 통상적인 블록 크기가 8KB이다. NTFS 파일 시스템에선 이를 클러스터라고 표현한다. 파일은 데이터의 저장 단위로서 전자 문서의 일 형태이다.

[0003] 여기서, 블록 사이즈 이하의 파일 크기를 갖는 파일이 파일 시스템에 의해 저장될 경우, 디스크의 비효율적인 사용이 발생된다. 예를 들면, 2KB의 파일이 상기 8KB의 블록 크기의 디스크에 저장되면, 6KB, 75%의 빈 공간이 남게 되어 디스크가 비효율적으로 사용된다. 블록 크기가 클수록 Read/Write 속도가 빨라지므로 공간 낭비를 줄이자고 무작정 블록 크기를 줄일 수도 없다.

[0004] 파일 시스템에서는 기본적인 파일 관리의 서비스만 제공되므로, 사용자가 스스로 파일의 접근 권한을 관리해야 한다. 또한, 사용자가 파일의 백업을 관리해야 한다. 기본적인 파일 관리의 서비스 이외에 파일의 접근, 백업 등을 관리하기 위해서는 사용자가 별도의 전용 하드웨어 / 소프트웨어의 장비를 사용해야 하므로, 장비 도입 / 유지에 많은 비용이 발생된다.

[0005] 한편, 블록 체인은 클라우드 기술 및 P2P 기술에 기반된 데이터의 기록 기술로서 누구나가 데이터의 저장 및 관리의 주체가 되어 동일한 정보를 각각 기록하는 기술이다. 여기서, 블록 체인의 "블록"은 데이터 저장 단위를 의미하는 것으로서, 파일 시스템의 "블록"이 디스크에서 하나의 파일이 할당될 수 있는 최소의 공간이라는 의미와 차이가 있다.

[0006] 블록 체인의 기술은 상기 파일 시스템에 기반된 파일 단위로 저장하는 것이 아니라, 분산 데이터베이스 환경에 따른 블록 체인의 DB에 데이터를 블록 단위로 저장한다. 블록 단위는 OS에 따라 종속되는 파일 시스템의 상기 블록 사이즈에 대한 제한을 받지 않는다. 즉, 블록 체인의 1개 단위의 저장 영역에 복수개의 데이터 단위가 저장될 수 있고, 8K의 블록 사이즈의 제한을 받지 않고 파일의 크기 2K만큼만 저장 공간을 사용하고, 상기의 6KB의 빈 공간은 남는 공간으로 활용되어 다른 파일이 저장되는데 사용된다.

[0007] 블록 체인의 기술은 데이터의 저장 단위인 블록에 대해 접근 가능 권한이 있는 경우에 한하여 데이터 접근을 허락하므로 보안성을 제공한다.

[0008] 블록 체인의 기술은 블록의 저장 주체인 노드들간에 동일 데이터가 저장된 블록의 기록을 동기화하므로 백업을 제공한다.

[0009] 또한, 블록 체인의 기술은 범용 컴퓨터 장비에 적용이 가능하므로 고가의 하드웨어 및 소프트웨어의 장비없이 비용이 절감되는 구축을 제공한다.

**선행기술문헌**

**특허문헌**

[0010] (특허문헌 0001) 한국등록특허 10-1637868(2016.07.01)

**발명의 내용**

**해결하려는 과제**

[0011] 본 발명은 상기와 같은 종래 기술의 인식하에 창출된 것으로서, 블록 체인 기술을 이용하여 파일 시스템의 블록 사이즈의 제한없이 데이터의 저장 공간을 효율적으로 사용하는 장치 및 방법을 제공하는 것을 목적으로 한다.

- [0012] 본 발명의 다른 목적은 장치가 누구나 접근 가능한 공개 장부의 서비스를 제공하는 것이 아니라 권한없는 사용자들에게 데이터의 접근이 허락되지 않고 서비스 제공자만 데이터에 접근할 수 있는 모드가 설정되는 서비스를 제공하는데 있다.
- [0013] 본 발명의 다른 목적은 장치가 공개 장부의 서비스의 수수료를 제거하여 수수료가 없이 데이터를 관리하는데 있다.
- [0014] 본 발명의 다른 목적은 Private 모드 및 제거된 수수료를 기반으로 장치가 마이닝 노드의 작업 증명 과정을 단순화하여 데이터의 유효성 검증을 수행하는데 있다.
- [0015] 본 발명의 또 다른 목적은 단순화된 마이닝 과정에 의해 빨라지는 블록의 생성 속도를 지연시키고자 딜레이를 주어 블록 생성 시간을 조절하고, 보관 대상의 트랜잭션이 발생된 경우에만 블록을 생성하는데 있다.

**과제의 해결 수단**

- [0016] 일 측면에 따른, 복수의 노드 단말들이 데이터를 동기화하여 저장하기 위해 형성한 블록 체인 및 상기 블록 체인의 데이터의 저장, 조회 및 삭제를 관리하는 장치는, 조회 요청에 의해 제공된 체인 정보를 이용하여 상기 블록 체인을 관리하는 체인 명령 및 조회 요청에 의해 제공된 매핑 정보를 이용하여 상기 데이터를 관리하는 데이터 명령을 상기 노드 단말로 전송하여 상기 데이터의 관리를 제어하는 메인 제어부; 및 상기 메인 제어부로부터 상기 체인 명령에 의해 생성된 체인의 상기 체인 정보를 입력받아 체인 테이블에 저장하고, 상기 요청에 따라 저장된 체인 정보를 조회하여 제공하고, 상기 메인 제어부로부터 상기 데이터 명령에 의해 저장된 데이터 및 트랜잭션의 매핑 정보를 입력받아 체인 테이블에 저장하고, 상기 요청에 따라 저장된 매핑 정보를 조회하여 제공하는 DB 관리부를 포함한다.
- [0017] 상기 블록 체인은, 권한이 정의된 사용자에게만 데이터 접근이 허락되는 Private 모드를 제공하는 제 1기능; 블록 체인의 데이터 처리에 수수료가 없는 제 2기능; 블록 체인에 요구되는 노드 단말에 대한 마이닝 노드의 작업 증명 과정에서 노드의 검증 처리를 제거하고 데이터의 유효성 검증 처리를 수행하는 제 3기능; 상기 작업 증명 과정에 의해 단축된 블록의 생성 시간을 지연시키기 위해 딜레이를 주어 블록 생성 시간을 지연시키는 제 4기능; 및 저장 대상의 트랜잭션이 발생된 경우에만 블록을 생성하는 제 5기능 중에서 적어도 하나 이상의 기능이 설정되어 각각의 노드 단말에서 실행된다.
- [0018] 상기 장치는, 웹 페이지를 통해 사용자 인터페이스를 제공하여 상기 체인 및 데이터의 관리 정보를 입력받고, 입력된 관리 정보의 처리 결과를 제공하는 웹 관리부를 더 포함한다.
- [0019] 상기 장치는, 상기 관리를 위한 스케줄링 정보를 등록받고, 등록된 스케줄링 정보에 대응되는 프로세스의 실행을 상기 메인 제어부로 요청하는 스케줄러부를 더 포함하고, 상기 메인 제어부는 상기 요청에 의해 대응되는 명령을 전송한다.
- [0020] 다른 측면에 따른, 상기 노드 단말은, 블록을 생성해서 체인에 연결하는 마이닝 처리 및 블록 체인의 데이터 정보를 API(Application program interface)를 통해 제공하는 마이닝 API 노드 단말; 및 마이닝 API 노드 단말에 의해 블록 체인의 데이터 관리가 동기화되는 데이터 동기화 노드 단말로 구성된다.
- [0021] 상기 마이닝 API 노드 단말은, 상기 체인 명령을 전송받고, 전송받은 체인 명령에 따른 블록 체인의 생성, 삭제 및 실행을 처리하고, 처리 결과를 응답하는 체인 생성부; 상기 데이터 명령을 전송받고, 전송받은 데이터 명령을 처리하고, 처리 결과를 응답하는 명령 프로세서부; 및 상기 처리에 따른 데이터 명령을 블록 체인에 대해 실행하는 블록 체인 처리부를 포함한다.
- [0022] 상기 데이터 동기화 노드 단말은, 상기 체인 명령을 전송받고, 전송받은 체인 명령에 따른 블록 체인의 생성, 삭제 및 실행을 처리하고, 처리 결과를 응답하는 체인 생성부; 및 상기 마이닝 API 노드 단말의 블록 체인 처리부와 포트 연결되고, 연결된 포트를 통해 전송된 데이터 명령을 동기화하여 실행하는 블록 체인 처리부를 포함한다.
- [0023] 다른 측면에 따르면, 상기 메인 제어부는, 블록 체인에 의해 관리되는 각각의 데이터 파일의 파일 정보로부터 체인 아이디를 생성하고, 생성된 체인 아이디에 수식을 적용하여 포트 아이디를 생성하고, 생성된 체인 아이디를 이용하여 블록 체인에 저장되는 파일의 폴더 경로를 생성한다.
- [0024] 상기 메인 제어부는, 저장 대상의 데이터 파일의 파일 정보를 입력받고, 입력된 파일 정보에 대응되는 체인 정보를 상기 체인 테이블로부터 조회하고, 조회된 체인 정보에 따른 상기 데이터 명령을 상기 노드 단말로 전송하



여 데이터 저장을 요청하고, 상기 데이터 명령의 저장 결과를 전송받고, 상기 DB 관리부는, 전송된 저장 결과의 파일 아이디, 블록 체인에 의해 저장시 생성된 트랜잭션 아이디 및 저장 시간을 포함하는 매핑 정보를 상기 매핑 테이블에 저장한다.

[0025] 상기 메인 제어부는, 파일 정보를 이용하여 생성된 체인 아이디를 키로 하여 상기 체인 테이블로부터 체인 정보를 조회하는데 실패할 경우, 생성된 체인 아이디를 갖는 블록 체인의 생성 및 실행을 요청하는 상기 체인 명령을 전송하고, 상기 체인 명령의 실행 결과를 전송받고, 상기 DB 관리부는, 전송된 실행 결과를 이용하여 각 체인별로 체인 아이디, 루트 디렉토리 및 실행 상태를 포함하는 상기 체인 정보를 생성하고, 생성된 체인 정보를 상기 체인 테이블에 저장한다.

[0026] 상기 메인 제어부는, 조회 대상의 데이터 파일의 파일 정보를 입력받고, 입력된 파일 정보에 대응되는 매핑 정보를 상기 매핑 테이블로부터 조회하고, 조회된 매핑 정보에 따른 상기 데이터 명령을 상기 노드 단말로 전송하여 데이터 조회를 요청하고, 조회된 파일 및 상기 데이터 명령의 조회 결과를 수신하고, 전송된 파일을 제공하는 것을 특징으로 하는 장치.

[0027] 또 다른 측면에 따른, 복수의 노드 단말들이 데이터를 동기화하여 저장하기 위해 형성한 블록 체인 및 상기 블록 체인의 데이터를 관리하는 장치가 데이터 파일을 저장하는 방법은, 저장 대상의 데이터 파일의 파일 정보로부터 체인 아이디를 생성하는 단계; 생성된 체인 아이디를 키로 하여 체인 테이블로부터 체인 정보를 조회하는 단계; 조회된 체인 정보에 따라 상기 파일의 저장을 요청하는 데이터 명령을 생성하고, 상기 노드 단말로 생성된 데이터 명령 및 파일을 전송하여 상기 블록 체인에 저장하는 단계; 및 상기 노드 단말로부터 전송받은 파일의 저장 결과를 이용하여 매핑 테이블에 데이터 및 트랜잭션의 매핑 정보를 저장하는 단계를 포함한다.

[0028] 또 다른 측면에 따른, 복수의 노드 단말들이 데이터를 동기화하여 저장하기 위해 형성한 블록 체인 및 상기 블록 체인의 데이터를 관리하는 장치가 데이터 파일을 조회하는 방법은, 사용자 단말로부터 조회 대상의 파일의 아이디를 입력받는 단계; 입력된 파일 아이디를 키로 하여 매핑 테이블로부터 매핑 정보를 조회하는 단계; 조회된 매핑 정보에 따라 상기 파일의 조회를 요청하는 데이터 명령을 생성하고, 상기 노드 단말로 생성된 데이터 명령을 전송하는 단계; 상기 노드 단말로부터 조회된 데이터 파일을 수신하는 단계; 및 수신된 파일을 상기 사용자 단말로 전송하는 단계를 포함한다.

**발명의 효과**

[0029] 본 발명의 일 측면에 따르면, 파일 시스템의 블록 크기 제한에 따라 빈 공간이 제거된 블록 체인의 저장 영역에 데이터를 보관하므로 저장 영역에 대비하여 효율적인 데이터 저장이 가능하다.

[0030] 또한, 본 발명의 다른 측면에 따르면, 블록 체인에 설정된 Private 모드에 의해 권한이 설정된 사용자만 관리 서버를 통해 데이터의 액세스 및 관리가 가능하므로 데이터의 보안이 유지될 수 있다.

[0031] 또한, 본 발명의 다른 측면에 따르면, 노드 단말을 관리 서버로부터 명령을 수신하여 마이닝 기능 및 API 기능을 제공하는 명령 노드와 상기 명령 노드에 데이터가 동기되는 동기화 노드로 구성함으로써, 데이터 보안 및 데이터 백업을 제공할 수 있다.

[0032] 또한, 본 발명의 다른 측면에 따르면, 블록 체인의 트랜잭션 발생에 따른 공개 장부의 서비스의 수수료를 제거하여 수수료가 없이 데이터를 관리하므로 적은 비용으로 블록 체인 기반의 DB의 구축이 가능하다.

**도면의 간단한 설명**

[0033] 본 명세서에 첨부되는 다음의 도면들은 본 발명의 바람직한 실시예를 예시하는 것이며, 후술한 발명의 상세한 설명과 함께 본 발명의 기술사상을 더욱 이해시키는 역할을 하는 것이므로, 본 발명은 그러한 도면에 기재된 사항에만 한정되어 해석되지 않아야 한다.

도 1 내지 도 5는 본 발명이 적용되는 블록 체인의 예시도이다.

도 2는 도 1의 망 관리 시스템의 개략적 내부 구성도이다.

도 6은 본 발명의 블록 체인에 설정되는 고유 기능들의 개략적 구성도이다.

도 7은 본 발명의 일 실시예에 따른 데이터 관리 시스템의 개략적 내부 구성도이다.

도 8은 본 발명의 일 실시예에 따른 블록 체인에 데이터를 저장하는 방법의 개략적 순서도이다.

도 9는 본 발명의 일 실시예에 따른 데이터 조회 방법의 개략적 순서도이다.

**발명을 실시하기 위한 구체적인 내용**

- [0034] 이하, 첨부된 도면을 참조하여 본 발명의 바람직한 실시예를 상세히 설명하기로 한다. 이에 앞서, 본 명세서 및 청구 범위에 사용된 용어나 단어는 통상적이거나 사전적인 의미로 한정해서 해석되어서는 아니되며, 발명자는 그 자신의 발명을 가장 최선의 방법으로 설명하기 위해 용어의 개념을 적절하게 정의할 수 있다는 원칙에 입각하여 본 발명의 기술적 사상에 부합하는 의미와 개념으로 해석되어야만 한다. 따라서, 본 명세서에 기재된 실시예와 도면에 도시된 구성은 본 발명의 가장 바람직한 일 실시예에 불과할 뿐이고 본 발명의 기술적 사상에 모두 대변하는 것은 아니므로, 본 출원 시점에 있어서 이들을 대체할 수 있는 다양한 균등물과 변형예들이 있을 수 있음을 이해하여야 한다.
- [0035] 도 1은 본 발명이 적용되는 블록 체인(100)의 예시도이다.
- [0036] 블록 체인(100)은 클라우드 저장 기능과 P2P 기능을 기반으로 구축된 분산형 데이터베이스이다. 대표적으로, 비트코인, 이더리움 등은 블록 체인을 기반으로 제공된다. 블록 체인(100)은 데이터가 저장되는 단위의 블록(101)들과 블록 사이의 데이터 링크에 해당되는 체인(102)으로 구성된다.
- [0037] 블록 체인(100)은 분산형/관계형 데이터 DB로서 파일 시스템으로부터 독립되어 DB에 할당된 저장 영역을 하나의 영역으로 하여 데이터를 저장한다. 따라서, 파일 시스템에 종속되는 블록 크기의 제한을 받지 않는다. 따라서, 상기 블록 크기의 제한에 의해 발생하는 빈 영역이 발생되지 않는다.
- [0038] 도 2는 도 1의 블록 체인(100)이 구축되는 노드 단말(200)의 예시도이다.
- [0039] 상기 노드 단말(200)은 데이터의 저장 공간을 제공하는 컴퓨터 장치이다. 노드 단말(200)은 스마트 단말, 노트북, PC, 서버 등으로 네트워크 기능, 저장 기능 및 프로세싱 기능을 갖기만 하면 스펙에 제한을 두지 않는다. 블록 체인에서는 상기 컴퓨터 장치를 노드라 한다.
- [0040] 도 3은 도 2의 노드 단말(200)에 4개의 고유한 체인(300)이 설치된 것을 예시한다.
- [0041] 공개 장부 기술의 특성상 한 개의 노드 단말(200)에는 한 개의 체인이 수행되는 것이 보편적인 방법이다. 화폐 거래의 장부 기록이 데이터의 신뢰성을 보장받기 위해 한 개의 공개 장부를 두고, 각 노드들이 동기성 및 공개성을 기반으로 동일한 공개 장부를 1개의 체인으로 보관하여 관리하기 때문이다.
- [0042] 하지만, 본 발명은 공개 장부를 보관하는 것이 아닌 지속적으로 증가하는 문서, 이미지 등의 데이터의 관리 및 분류가 목적이기 때문에 그 목적에 따라 분류된 복수개의 체인에 기반된 데이터의 분산화가 바람직하다.
- [0043] 도 4는 3개의 노드 단말(200)이 4개의 체인에 대해 페어링을 통해 동기화되는 예시도이다.
- [0044] 각 노드 단말(200)은 #1 ~ #4의 4개의 고유한 블록 체인을 갖는다. 각각의 체인은 페어링(401~404)을 맺고 데이터가 동기화된다. 각 체인은 3개의 노드로 구성되어 있다고 표현할 수 있다. 즉, 3개의 노드 단말(200)에 보관된 각각의 체인 #1은 제 1페어링(401)에 의해 동일한 데이터를 저장하도록 동기화된다. 동기화에 의해, 제 1노드의 체인 #1에 데이터의 저장, 삭제 등의 처리가 발생되면, 제 1페어링(401)을 맺은 나머지 제 2노드 및 제 3노드의 체인 #1도 동일한 데이터의 처리가 발생된다.
- [0045] 기술적으로, 페어링을 맺는 체인끼리는 동일한 포트를 사용하여 통신을 하게 된다. #1 ~ #4의 체인은 각각 고유한 포트를 사용함으로써 독립된 체인으로 동작할 수 있다.
- [0046] 도 5는 도 4의 블록 체인 환경에서 운영되는 복수개의 노드 그룹의 예시도이다.
- [0047] 노드 단말(200)은 3개의 노드로 구성된 노드 그룹 #1(500) 및 노드 그룹 #2(501)로 구성된다. 노드 그룹 #1(500)은 각 노드마다 체인 #1 ~ #4를 보유하고, 각각의 체인은 페어링된다. 노드 그룹 #2(501)은 각 노드마다 체인 #5 ~ #8을 보유하고, 각각의 체인은 페어링된다. 각 체인간의 페어링 연결은 간략한 표시를 위해 생략된다.
- [0048] 한 개의 노드가 #1 ~ #8의 8개 체인을 갖는 것과 비교하면, 분리된 노드 그룹에 의해 부하가 분산될 수 있다. 1개의 체인이 동작하기 위해선 CPU, 메모리, 저장 공간 등의 자원이 필요하게 되는데, 여러 개의 체인을 동시에 실행시키기엔 1개의 노드로 부하가 집중되므로, 노드 그룹으로 분리하여 부하가 분산될 수 있다. 또한, 부하의 분산을 통해, 그룹 노드의 1개의 노드가 물리적으로 고장이 발생할 경우에도 서비스 운용에 미치는 영향을 감소시킬 수 있다. 만약, 8개의 체인을 저장하는 노드가 고장일 경우, 8개의 체인에 대한 데이터 액세스가 불가능

된다.

- [0049] 도 6은 본 발명의 블록 체인의 위해 설정되는 수정 기능들의 개략적 구성도이다.
- [0050] 본 발명은 블록 체인의 고유한 기능을 적용받는다. 첫 번째로, 블록 체인의 DB를 이용하므로 저장 영역의 공간 낭비가 제거된다. 두 번째로, 데이터가 블록 체인의 블록에 포함되므로 이 데이터에 접근하기 위해선 특별한 권한(관리자 계정)을 획득해야만 하므로 보안성이 강화된다. 세 번째로, 블록 체인이 보유한 노드들 간에 동기화가 지원되어 실시간 백업이 가능하여 원본 손실의 위험이 회피된다. 마지막으로, 블록체인 기술을 이용한 시스템을 구축하기 위해 고가의 장비나 소프트웨어 없이도 x86 서버와 오픈 소스를 이용해서도 구현이 가능하므로 비용이 절감된다.
- [0051] 이때, 비트 코인이나 이더리움 등 블록 체인 기술을 대표하는 종래의 기술을 바로 적용할 수는 없다. 본 발명을 구현하는 데에는 최적화된 별도의 체인이 요구된다. 물론, 종래의 기술을 수정 및 응용하여 본 발명의 블록 체인을 구현하는 것도 가능하다. 바람직하게, 본 발명에서는 이더리움의 블록 체인으로부터 응용될 수 있다. 비트 코인의 경우, 화폐의 유통, 사용이라는 목적에만 최적화되어 있는 기술인데다 이더리움처럼 스마트 컨트랙트 같은 개념도 구현되지 않은 기술이라 본 발명에서 제시하는 일반 데이터 저장에는 적합하지 않다. 물론, 이더리움의 블록 체인 방식도 그대로 사용 가능한 것은 아니며 아래에서 제시한 제 1기능 내지 제 5기능의 수정 기능이 필요하다.
- [0052] 제 1기능에 의해, 블록 체인의 노드들 사이의 네트워크 구성은 Private 모드로 설정된다. 권한 있는 사용자만 사용할 서비스이기 때문에 타 사용자들에게 공개될 필요는 없다.
- [0053] 제 2기능에 의해, 블록 체인의 데이터 처리에 수수료가 없는 것으로 설정된다. 블록 체인이란 기술은 본래 탈중앙 집권을 표방한 공공의 데이터베이스이다. 따라서, 블록 체인이라는 공공의 데이터베이스를 사용하기 위해선 수수료(사용료)가 수반된다. 하지만, 본 발명은 공공의 데이터베이스가 아닌 사용자만의 전용 데이터베이스로 운용되므로 수수료의 개념이 불필요하다. 따라서 코드 수정을 통해 수수료가 발생되지 않도록 설정되어야 한다.
- [0054] 제 3기능에 의해, 마이닝 노드의 POW (작업증명, Proof of Work) 과정이 단순화되도록 설정된다. 공공의 블록 체인이라면 누구나 자발적으로 체인의 노드로 참여하는 것을 유도하기 위해 마이닝이라는 과정을 수행한 노드에 금전적인 이득을 제공하고 있다. 이 마이닝 과정은 HW 자원을 최대한으로 사용하도록 되어 있는데 이 역시 본 발명이 적용할 서비스의 형태엔 불필요한 과정이다. 따라서, 마이닝 과정 중 데이터의 유효성 검증 부분만 남기고, 랜덤 해시 계산에 의한 nonce 값 찾기와 같은 과정은 제거됨으로써 블록이 빠르게 생성되고 CPU / 메모리 등의 자원은 최소화되도록 설정된다.
- [0055] 제 4기능에 의해, 제 3기능의 단순화된 마이닝 과정에서 블록의 생성 속도가 불필요하게 빨라지게 되므로 적절한 속도로 블록이 생성될 수 있도록 딜레이가 설정된다. 딜레이 설정에 의해 블록의 생성 속도가 지연된다.
- [0056] 제 5기능에 의해, 종래의 블록 체인은 트랜잭션이 발생하지 않은 상태에서도 빈 블록이 끊임없이 생성되나 본 발명에서는 불필요한 오버헤드이므로 처리할 트랜잭션이 있을 경우에만 블록 생성을 진행하도록 설정된다.
- [0057] 상기의 제 1기능 내지 제 5기능의 수정 기능이 설정된 블록 체인이 본 발명에 적용된다. 이하에서는, 설명의 편의를 위해 신용카드사가 고객의 카드 결제시 발생하는 서명 이미지의 데이터 파일을 본 발명의 블록 체인에 저장하는 것으로 예시한다. 물론, 본 발명은 데이터 파일의 내용 및 포맷에 제한을 두지 않는다.
- [0058] 신용카드사는 POS 단말기로부터 VAN사를 거쳐 전달되는 고객의 서명 이미지를 개개의 파일로 저장하는데 각각의 승인 건마다 VAN사에서 부여하는 일련번호가 할당되고, 이 일련번호를 파일명으로 하여 압축되지 않은 흑백 비트맵 파일(예 : 일련번호.bmp)로 저장한다. 일련번호는 카드 결제의 발생 시각을 포함하는 고유 정보라 가정한다. 서명 이미지의 파일은 2KB 내외로 파일 시스템에 의해 저장될 경우, 블록 사이즈의 제한을 받아 6KB의 빈 영역이 남게 되어 비효율적이다.
- [0059] 여기서, 하루에 500만 건의 2KB의 서명 이미지 파일이 생성되고, 법 적용을 받아 5년간 보관되는 것으로 가정한다. 5년간 저장되는 파일의 개수는  $500만 * 365 * 5 = 9,125,000,000$ 개가 된다. 파일의 크기는  $9,125,000,000 * 2 * 1024$ 에 의해, 16TB가 된다.
- [0060] 16TB를 1개의 체인으로 구성하여 1개의 노드에 저장하는 것은 부하가 집중되어 비효율적이다. 때문에, 복수개의 체인을 구성하여 데이터 크기를 줄이는 것이 효율적이다. 본 발명에서는 저장될 데이터가 발생하는 특성을 고려하여 체인의 관리 단위를 정의하는 것이 바람직하다. 본 발명에서는 서명 이미지의 월 단위로 체인이 생성되고,

5년간 보관되는 체인 60개가 운영되고, 5년이 지난 체인은 1개월이 지날 때마다 삭제되는 것으로 운영된다.

- [0061] 참고로, 블록 체인에 데이터가 저장되는 방식은 '키/밸류' 형식으로 저장된다. 서명 이미지의 파일의 경우, 파일명이 고유하므로 "키"로 활용될 수 있고, 이미지 파일은 "밸류"에 해당된다.
- [0062] 도 7은 본 발명의 일 실시예에 따른 데이터 관리 시스템(700)의 개략적 내부 구성도이다.
- [0063] 본 발명의 일 실시예에 따른 데이터 관리 시스템(700)은 관리 서버(710), 마이닝 API 노드 단말(730) 및 데이터 동기화 노드 단말(750)을 포함하여 구성된다.
- [0064] 상기 관리 서버(710)는 웹 관리부(711), 스케줄러부(713), 메인 제어부(715) 및 DB 관리부(717)를 포함하여 구성된다.
- [0065] 상기 웹 관리부(711)는 웹 사이트에 접속한 사용자 단말을 상대로 사용자 UI를 이용하여 데이터 관리 서비스를 제공한다. 웹 관리부(711)는 관리 서비스를 제공하여 체인의 현재 상태, 체인에 담긴 레코드의 현황, 체인으로부터 조회된 데이터를 사용자 단말로 제공한다.
- [0066] 여기서, "데이터 관리 서비스"는 체인 및 데이터 각각에 대한 생성, 조회, 수정 및 삭제를 포함한다. 웹 관리부(711)는 웹 사이트에 접속한 사용자에게 사용자 UI를 제공하고, 사용자 UI를 통해 데이터 관리 요청을 수신하고, 수신된 사용자의 요청을 메인 제어부(715)로 전달한다. 메인 제어부(715)에 의해 사용자의 요청이 처리되면, 웹 관리부(711)는 메인 제어부(715)로부터 처리 결과를 수신하고, 사용자 UI를 통해 수신된 처리 결과를 사용자에게 제공한다. 물론, 처리 결과는 체인 및 데이터에 대해 생성, 조회, 수정 및 삭제가 실행된 결과이다.
- [0067] 상기 스케줄러부(713)는 데이터 관리의 스케줄링 정보를 등록받고, 등록된 스케줄링 정보의 프로세스의 실행을 메인 제어부(715)로 요청한다. 메인 제어부(715)에 의해 스케줄러부(713)의 요청이 처리되면, 스케줄러부(713)는 메인 제어부(715)로부터 스케줄링된 프로세스의 처리 결과를 수신한다.
- [0068] 웹 관리부(711)가 사용자로부터 데이터 관리를 예약받으면, 스케줄러부(713)에 사용자의 예약 정보가 스케줄링 정보로서 등록된다. 웹 관리부(711)는 사용자로부터 실시간 또는 스케줄링의 방식에 따라 관리 요청을 수신할 수 있다. 등록된 스케줄링 정보는 사용자에게 의해 예약된 시간 및 예약된 관리 정보를 포함한다. 예를 들어, 매월 생성된 체인이 5년간 데이터를 저장한 후, 1개월이 경과될 때마다 삭제되는 프로세스가 스케줄링 정보로서 등록될 수 있다.
- [0069] 상기 메인 제어부(715)는 노드로 상기 블록 체인을 관리하는 체인 명령 및 상기 데이터를 관리하는 데이터 명령을 전송한다. 상기 전송에 의해, 블록 체인 및 데이터의 관리가 제어된다. 메인 제어부(715)는 관리 대상의 파일의 파일명으로부터 설정된 규칙에 따라 체인 아이디 및 파일 아이디를 생성한다.
- [0070] 저장의 경우, 메인 제어부(715)는 저장 대상의 파일명으로부터 체인 아이디를 생성하고, 생성된 체인 아이디의 체인 정보를 조회한다. 조회된 체인 정보로부터 파일의 경로 정보를 식별하고, 식별된 경로 정보에 데이터가 저장될 것을 요구하는 데이터 명령 및 파일을 노드로 전송한다. 그러면, 노드는 경로 정보에 따라 대응되는 체인 및 경로에 파일을 저장한다.
- [0071] 조회의 경우, 메인 제어부(715)는 조회 대상의 파일명으로부터 대응되는 매핑 정보를 조회한다. 조회된 매핑 정보의 트랜잭션 아이디가 포함되는 데이터 조회의 데이터 명령을 노드로 전송한다. 그러면, 노드는 트랜잭션 아이디에 따라 대응되는 파일을 조회하고, 조회된 파일을 메인 제어부(715)로 전송한다. 메인 제어부(715)는 전송 받은 파일을 웹 관리부(711)를 통해 사용자 단말로 제공한다.
- [0072] 삭제의 경우, 조회의 데이터 명령 처리를 따르고, 노드는 트랜잭션 아이디에 따라 대응되는 파일을 삭제한다.
- [0073] 상기 DB 관리부(717)는 생성된 체인의 체인 정보가 체인 테이블에서 저장, 조회 및 삭제되는 것을 관리한다. DB 관리부(717)는 메인 제어부(715)를 통해 수신된 체인 생성의 실행 결과를 이용하여 각 체인별로 "체인 아이디, 루트 디렉토리 및 실행 상태"를 포함하는 체인 정보를 생성하고, 생성된 체인 정보를 체인 테이블에 저장한다. 체인 아이디는 체인을 고유하게 식별하는 정보이다. 루트 디렉토리는 체인이 저장되는 루트 디렉토리이다. 체인 아이디 및 루트 디렉토리는 파일에 대응되는 체인을 식별하는 경로 정보를 형성한다. 실행 상태는 관리 명령의 실행을 위해 파일이 저장되는 체인이 실행 상태가 아닐 경우, 체인의 실행을 위해 참조된다.
- [0074] 이후, 파일의 저장 요청이 발생되면, DB 관리부(717)는 메인 제어부(715)의 조회 요청에 따라 관리 대상의 파일에 대응되는 체인 정보를 체인 테이블로부터 조회한다. 조회된 체인 정보로부터 파일에 대응되는 체인 및 접근

경로의 정보가 식별되고, 메인 제어부(715)는 식별된 정보를 이용하여 파일의 저장을 위한 데이터 명령을 내릴 수 있다.

- [0075] 또한, 상기 DB 관리부(717)는 저장이 완료된 데이터 및 저장 트랜잭션의 매핑 정보가 매핑 테이블에서 저장, 조회 및 삭제되는 것을 관리한다. DB 관리부(717)는 메인 제어부(715)를 통해 수신된 데이터 저장의 저장 결과를 이용하여 파일 아이디, 블록 체인에 의해 저장시 생성된 "파일 아이디, 트랜잭션 아이디 및 저장 시간"을 포함하는 매핑 정보를 매핑 테이블에 저장한다. 파일 아이디는 파일명으로부터 생성되고 파일을 식별하는 고유 키이다. 트랜잭션 아이디는 파일의 저장에 의해 블록 체인이 고유하게 생성하고 파일에 액세스하는 고유 정보이다. 저장 시간은 트랜잭션의 처리 시간이다.
- [0076] 이후, 저장된 파일의 관리(예 : 조회, 삭제 등) 요청이 발생되면, DB 관리부(717)는 메인 제어부(715)의 조회 요청에 따라 관리 대상의 파일에 대응되는 매핑 정보를 매핑 테이블로부터 조회한다. 조회된 매핑 정보로부터 파일에 대응되는 트랜잭션 정보가 식별되고, 메인 제어부(715)는 식별된 트랜잭션 정보를 이용하여 저장된 파일의 관리를 위한 데이터 명령을 내릴 수 있다.
- [0077] 관리 서버(710)는 WAS 포트 통신으로 연결된 마이닝 API 노드 단말(730) 및 데이터 동기화 노드 단말(750)에 대해 블록 체인 및 데이터의 관리를 제어한다. 노드 단말(200)은 마이닝 API 노드 단말(730) 및 데이터 동기화 노드 단말(750)에 해당된다.
- [0078] 상기 마이닝 API 노드 단말(730)은 WAS 포트 통신을 통해 메인 제어부(715)로부터 각종 명령을 수신하고, 수신된 명령에 따라 블록을 생성해서 체인에 연결하는 Mining 작업과, 관리 명령(예 : 저장 / 조회 등)을 수신하고, 수신된 명령을 처리하고, 처리 결과를 메인 제어부(715)로 응답하는 API 작업을 수행한다. 마이닝 API 노드 단말(730)은 체인 생성부(731), 블록 체인 모듈(733), 명령 프로세서부(735), 블록 체인 처리부(737)를 포함하여 구성된다.
- [0079] 상기 체인 생성부(731)는 메인 제어부(715)로부터 체인을 관리하는 체인 명령을 수신하고, 수신된 체인 명령에 대응되는 체인의 생성, 실행(구동) 및 삭제 처리를 수행한다. 블록 체인의 "Chain Maker"는 체인 생성부(731)이다.
- [0080] 여기서, 노드에 체인이 만들어지기 위해선 체인의 실행 파일과 부수적인 설정 파일들이 각 노드의 약속된 디렉토리에 미리 배포되어 있어야 한다. 체인 생성부(731)는 배포된 실행 파일들과 설정 파일들을 포함하는 블록 체인 모듈(733)을 이용하여 체인 명령을 처리한다. 체인 명령이 삭제인 경우, 실행 중인 프로세서가 중지되고, 체인 운용으로 인해 생성된 파일들이 삭제된다.
- [0081] 상기 명령 프로세서부(command processor)(735)는 메인 제어부(715)로부터 데이터를 관리하는 데이터 명령을 수신하고, 수신된 데이터 명령에 대응되는 데이터의 저장, 조회 및 삭제 처리를 수행한다. 명령 프로세서부(735)는 메인 제어부(715)로부터 수신된 데이터 명령을 블록체인 프로세스(737)로 전달하는 역할을 한다. 한편, 명령 프로세서부(735)는 외부 서비스에 연결되어 데이터 명령의 수신이 가능하지만, 보안 연결이 요구된다.
- [0082] 상기 블록 체인 처리부(737)는 명령 프로세서부(735)로부터 데이터 명령을 수신하여 데이터 관리를 수행한다. 데이터 관리의 수행에 의해 데이터가 체인에 저장, 조회 및 삭제된다. 또한, 블록 체인 처리부(737)는 페어링 기능에 의해 타 노드의 블록 체인 처리부(757)와 동기화 포트 통신을 통해 동기된다.
- [0083] 상기 데이터 동기화 노드 단말(750)은 체인 생성부(751), 블록 체인 모듈(753) 및 블록 체인 처리부(757)를 포함하여 구성된다. 체인 생성부(751) 및 블록 체인 모듈(753)은 마이닝 API 노드 단말(730)의 구성과 동일하다. 또한, 블록 체인 처리부(757)는 마이닝 API 노드 단말(730)의 블록 체인 처리부(737)와 동기된다.
- [0084] 상기 Mining 작업과 API 작업은 마이닝 API 노드 단말(730)에서만 수행되는 것이 차이점이다. 물론, 모든 노드에 명령 프로세서부(735)를 두어 시스템 Fail-Over에 대응하도록 구성될 수도 있다. 다만, 본 발명의 실시예에서는 관리 서버(710)에 대한 마이닝 작업과 API 작업의 창구를 마이닝 API 노드 단말(730)로 일원화하여 데이터의 보안을 유지하고, 마이닝 API 노드 단말(730)과 페어링에 의해 데이터 동기화 노드 단말(750)이 동기화되어 데이터 백업을 제공하고자 한다. 즉, 데이터 보안의 측면에서 권한있는 사용자만 관리 서버(710)에 액세스성이 가능하고, 권한있는 관리 서버(710)만 마이닝 API 노드 단말(730)로 액세스성이 가능하므로 외부의 침입이 원천적으로 차단된다.
- [0085] 여기서, 메인 제어부(715) 및 노드 단말들(730, 750) 사이의 상기 WAS 포트 통신은 Java Servlet 형태로 구현되었다고 가정하면, Tomcat, JBOSS 등의 Web Application Server)에서 사용하는 포트를 통해 이루어지게 된다.

- [0086] 명령 프로세서부(735) 및 블록 체인 처리부(737) 사이는 RPC (Remote Procedure Call) 포트가 사용된다. 블록 체인 처리부들(737, 750) 사이의 동기화를 위해선 Sync 포트가 사용된다. Sync 포트와 RPC 포트의 아이디는 체인 ID를 통해 계산될 수 있다.
- [0087] 도 8은 본 발명의 일 실시예에 따른 블록 체인에 데이터를 저장하는 방법의 개략적 순서도이다.
- [0088] 관리 서버(710)에서 파일 저장의 이벤트가 발생한다. 파일 저장은 사용자가 웹 관리부(711)의 웹 사이트를 통해 파일 저장을 요청할 수 있다. 이외에도 배치 처리 또는 스케줄러부(713) 등에 의해서도 파일 저장의 이벤트가 발생할 수 있다. 저장 대상의 파일이 웹 사이트의 지정된 디렉토리에 저장되면, 메인 제어부(715)가 이벤트 발생의 알림을 수신한다(S801).
- [0089] 메인 제어부(715)는 서비스 정책에 따라 파일의 파일 정보(예 : 파일명, 파일 시간 등)를 이용하여 체인 아이디를 생성한다(S811). 여기서, 파일 정보로부터 체인 아이디, 고유 키, 포트 번호 등이 생성될 수 있다. 생성 규칙은 파일명으로부터 고유 값의 추출, 변환, 계산 등의 처리가 수반되는 것이며 특별한 제한을 두지 않는다.
- [0090] 상기 서명 이미지의 경우, 날짜 시간의 파일명으로부터 년(2)+월(2)의 체인 아이디를 추출하여 생성한다. 예를 들어, 17년 1월에 생성된 서명 이미지 파일은 파일명에 '1701'이 포함되고, 파일명으로부터 추출된 '1701'의 체인 아이디가 생성된다. 고유 키는 파일명이 고유할 경우, 파일명을 그대로 사용할 수 있고, 상기 생성 규칙을 적용하는 것도 가능하다. 상기 포트 번호는 아래의 수식을 이용하여 생성된다.
- [0091] <수식>
- [0092] 동기화 포트 = 체인 아이디 + 10000
- [0093] RPC 포트 = 동기화 포트 + 3000
- [0094] 그러면, '1701' 체인 아이디의 경우, 블록 체인 처리부들(737, 757)의 동기화 포트는 '11701'이고, 명령 프로세서부(735)의 RPC 포트는 '14701'이다.
- [0095] 체인 아이디가 생성되면, 메인 제어부(715)는 DB 관리부(717)로 체인 아이디를 입력하여 체인 정보를 조회한다(S821). DB 관리부(717)는 체인 아이디를 키로 하여 체인 테이블을 조회하고, 체인의 체인 아이디, 루트 디렉토리 및 실행 상태를 포함하는 체인 정보를 조회하고, 조회된 체인 정보를 메인 제어부(715)로 제공한다.
- [0096] 만약, 체인 정보의 조회가 실패될 경우(S823), 메인 제어부(715)는 체인 아이디 및 루트 디렉토리를 포함하는 경로명이 포함되고 체인을 생성하는 체인 명령을 생성하고, 생성된 체인 명령을 각 노드의 체인 생성부들(731, 751)로 전송한다(S825). 체인 생성부들(731, 751)은 수신된 경로명에 따라 체인을 생성 및 실행한다. 체인의 생성이 완료되면, 체인 생성부들(731, 751)은 생성 결과를 메인 제어부(715)로 응답한다. 메인 제어부(715)는 응답된 생성 결과를 이용하여 DB 관리부(717)를 통해 체인 테이블에 체인 정보를 저장한다.
- [0097] 서명 이미지의 경우, 루트 디렉토리가 '/CHAIN\_DATA' 이고 체인 아이디가 '1701'이라면 데이터 디렉토리의 경로명은 정의된 서비스 정책에 따라 '/CHAIN\_DATA/datadir\_1701' 가 될 수 있다. 매월 단위로 발생된 서명 이미지가 대응되는 체인에 저장되는 경우, 매월 첫 번째 데이터의 저장이 발생될 경우, 체인을 생성하는 체인 명령이 실행된다. 만일, '/CHAIN\_DATA'의 체인 저장 공간이 부족해지면 사용자는 이 위치를 '/CHAIN\_DATA2'와 같이 다른 스토리지 영역으로 변경해 줌으로써 확장 이슈도 해결하는 것이 가능하다.
- [0098] 또한, 조회된 체인 정보의 실행 상태가 실행 중이 아니면(S827), 메인 제어부(715)는 체인 아이디 및 루트 디렉토리를 포함하는 경로명이 포함되고 체인을 실행하는 체인 명령을 생성하고, 생성된 체인 명령을 각 노드의 체인 생성부들(731, 751)로 전송한다(S829). 체인 생성부들(731, 751)은 수신된 경로명에 따라 체인을 실행한다. 체인의 실행에 의해 체인 테이블의 실행 상태는 실행 중으로 업데이트된다. 처리 효율을 위해 체인은 처리가 완료되면 실행이 종료될 수 있다.
- [0099] 실행 중인 체인의 체인 정보가 조회된 후, 메인 제어부(715)는 파일의 파일 정보(예 : 파일명, 파일 시간 등)로부터 고유한 파일 아이디를 생성한다. 생성된 파일 아이디는 파일을 식별하는 키에 해당된다. 파일 아이디의 생성을 위해, 파일 정보에 각종 변환, 해시 처리 등이 적용될 수 있다. 물론, 파일명이 고유할 경우, 메인 제어부(715)는 파일명을 파일 아이디로 할 수 있다. 메인 제어부(715)는 조회된 체인 정보 및 생성된 파일 아이디를 포함하는 데이터 명령을 생성하고, 생성된 데이터 명령 및 파일을 명령 프로세서부(735)로 전송하여 파일을 저장한다(S831).
- [0100] 여기서, 명령 프로세서부(735)는 수신된 데이터 명령을 블록 체인 처리부(737)로 전송하여 블록 체인에 데이터

가 저장되게 한다. 블록 체인에 저장되는 파일 정보는 키/밸류의 RDB 형식을 따르며, 파일과 관련된 기타 메타 정보(고유키 등)가 더 저장될 수 있다. 블록 체인 처리부(737)에 의해 데이터가 블록 체인에 저장되면, 동기화된 블록 체인 처리부(757)도 데이터 명령 및 파일을 동기화 포트를 통해 전송받고 파일을 저장하여 동기화 처리한다.

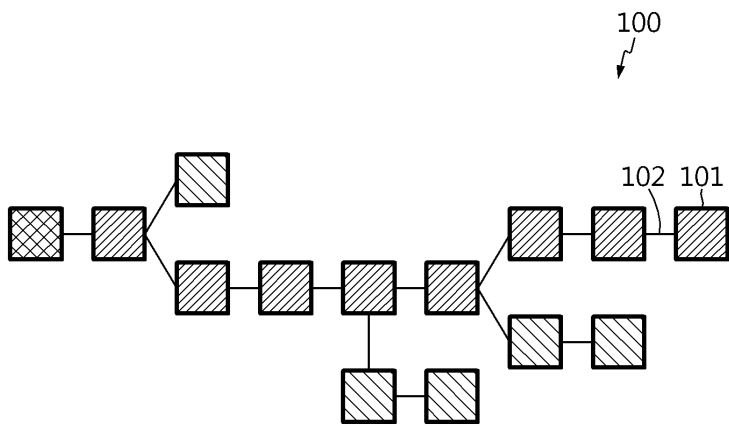
- [0101] 파일 저장이 완료되면, API 작업에 의해 명령 프로세서부(735)는 저장된 파일의 트랜잭션 아이디를 포함하는 저장 결과를 메인 제어부(715)로 응답한다. 트랜잭션 아이디는 블록 체인에 데이터 트랜잭션이 저장될 경우 고유하게 생성된다. 메인 제어부(715)는 응답된 저장 결과를 이용하여 DB 관리부(717)를 통해 매핑 테이블에 매핑 정보를 저장한다. 매핑 테이블에는 파일의 메타 정보가 더 저장될 수 있다.
- [0102] 상기 단계(S801) 내지 단계(S843)는 웹 사이트의 디렉토리에 저장된 파일들의 저장이 완료될 때까지 반복된다. 파일들의 저장이 완료된 후, 메인 제어부(715)는 저장이 완료된 파일들을 상기 디렉토리에서 삭제한다(S845).
- [0103] 이후, 메인 제어부(715)는 서비스 정책에 따라 조건이 만족된 체인을 체인 명령을 노드로 전송하여 삭제한다(S851). 서명 이미지의 경우, 5년 1개월이 경과된 체인이 노드로부터 자동 삭제된다.
- [0104] 도 9는 본 발명의 일 실시예에 따른 데이터 조회 방법의 개략적 순서도이다.
- [0105] 관리 서버(710)에서 파일 조회의 이벤트가 발생한다. 파일 조회는 웹 사이트에 접속된 사용자의 요청, 배치 처리 또는 스케줄러부(713) 등에 의해서 해당 이벤트가 발생할 수 있다. 메인 제어부(715)가 파일 조회의 이벤트가 발생되었음을 알리는 알림에 의해 파일 아이디를 수신한다(S901).
- [0106] 여기서, 메인 제어부(715)는 매핑 테이블의 매핑 정보를 조회하고, 조회된 매핑 정보를 웹 관리부(11)로 제공하여 사용자가 조회 대상의 파일을 선택하도록 검색 서비스를 제공하는 것도 가능하다.
- [0107] 메인 제어부(715)는 수신된 파일 아이디를 키로 하여 매핑 테이블로부터 매핑 정보를 조회한다(S903). 매핑 조회에 실패할 경우, 삭제 대상의 레코드가 존재하지 않는다는 적절한 오류 메시지를 제공한다. 또한, 서비스의 형태에 따라 특정 일자가 경과한 데이터일 경우, 제공을 거부하는 기능이 이 단계에서 추가되는 것도 가능하다.
- [0108] 매핑 정보가 조회되면, 메인 제어부(715)는 매핑 정보의 트랜잭션 아이디를 포함하고 데이터의 조회를 요청하는 데이터 명령을 생성하고(S911), 생성된 데이터 명령을 명령 프로세서부(735)로 전송하여 파일의 조회를 요청한다(S913).
- [0109] 여기서, 명령 프로세서부(735)는 수신된 데이터 명령을 블록 체인 처리부(737)로 전송하여 블록 체인에 저장된 파일의 조회를 요청한다. 블록 체인 처리부(737)에 의해 트랜잭션 아이디의 파일이 조회되고, 명령 프로세서부(735)는 조회된 파일 및 처리 결과를 API 작업에 따라 메인 제어부(715)로 전송한다. 메인 제어부(715)는 명령 프로세서부(735)로부터 파일을 다운로드 수신한다(S921). 다운로드 수신된 파일은 웹 관리부를 통해 사용자에게 다운로드 서비스된다(S931).
- [0110] 상기 단계(S901) 내지 단계(S931)는 조회 요청된 파일들의 조회가 완료될 때까지 반복될 수 있다. 파일들의 다운로드 서비스가 완료된 후, 메인 제어부(715)는 다운로드가 완료된 파일들을 다운로드 디렉토리에서 삭제한다
- [0111] 한편, 파일의 저장 및 조회 과정에서 데이터 보안을 위해 메인 제어부(715)는 암호화 및 복호화 처리를 수반하는 것이 가능하다. 또한, 웹 사이트의 서비스 정책에 따라 문서 파일이나 이미지 파일의 내용을 보여주는 서비스를 제공하는 것도 가능하다.
- [0112] 본 발명은 비록 한정된 실시예와 도면에 의해 설명되었으나, 본 발명은 이것에 의해 한정되지 않으며 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자에 의해 본 발명의 기술사상과 아래에 기재될 특허청구범위의 균등 범위 내에서 다양한 수정 및 변형이 가능함은 물론이다.

**부호의 설명**

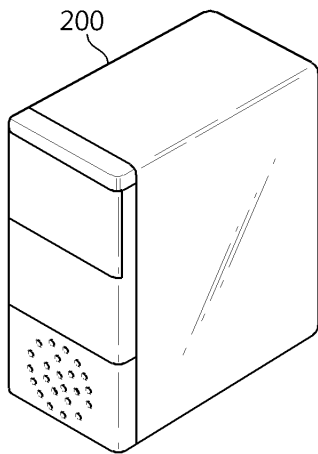
- [0113] 700 : 데이터 관리 시스템 :            710 : 관리 서버
- 711 : 웹 관리부                            713 : 스케줄러부
- 715 : 메인 제어부                            717 : DB 관리부
- 730 : 마이닝 API 노드 단말            750 : 데이터 동기화 노드 단말

도면

도면1

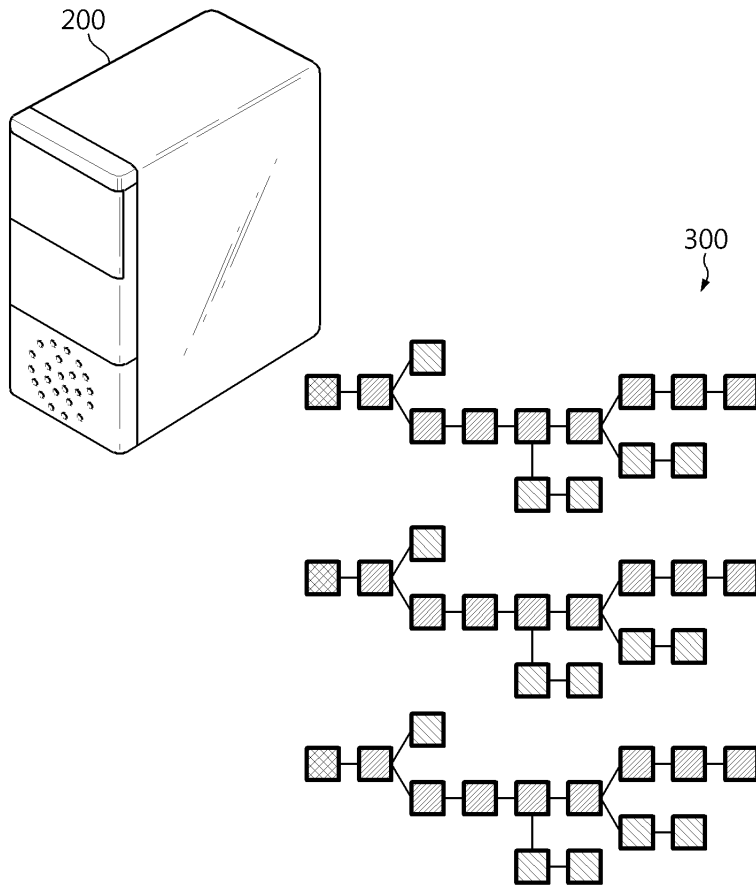


도면2

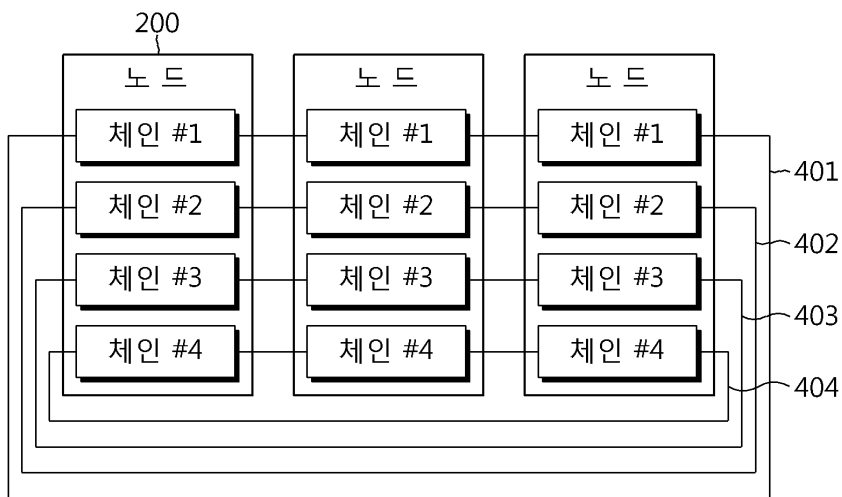




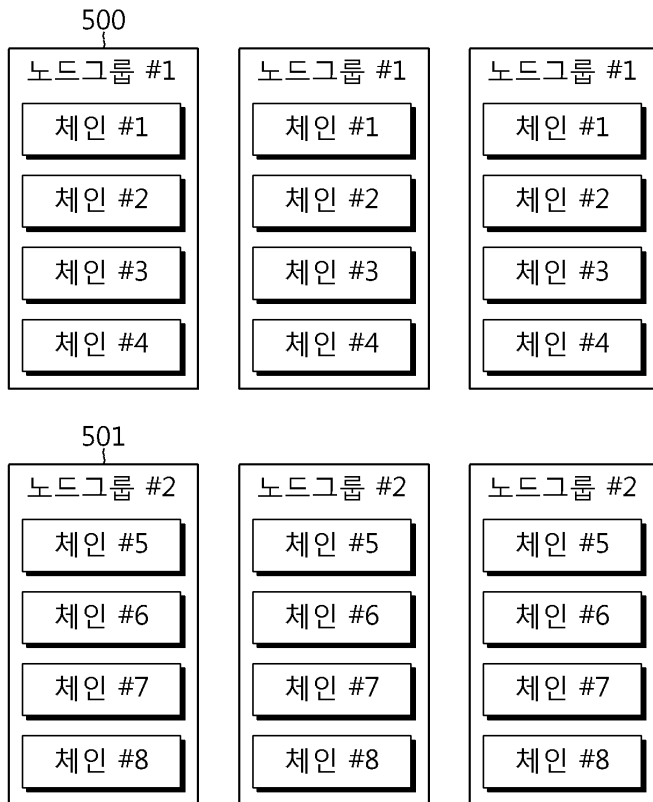
도면3



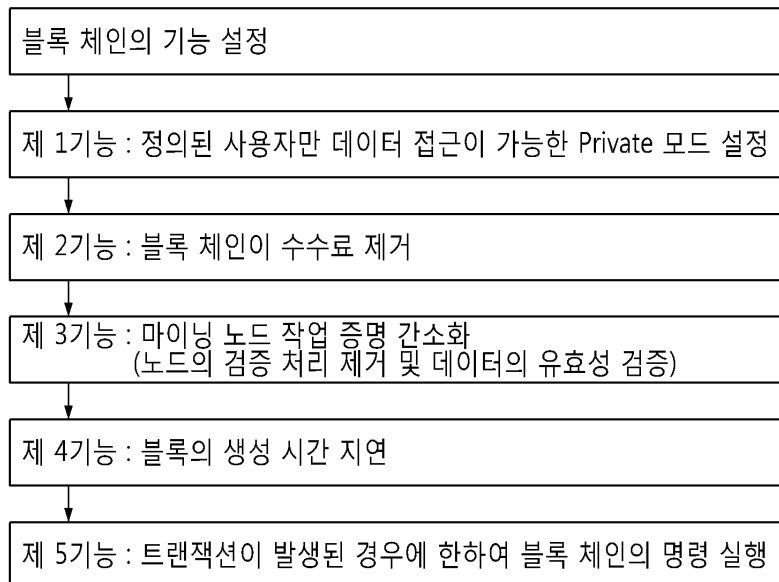
도면4



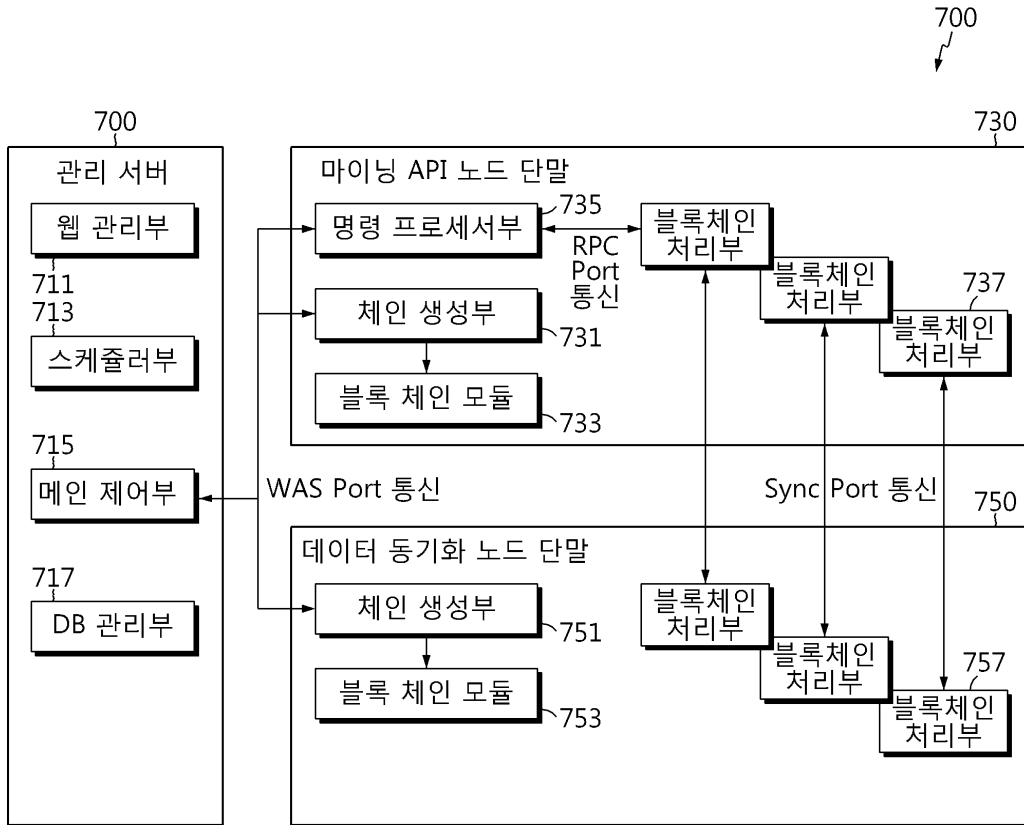
도면5



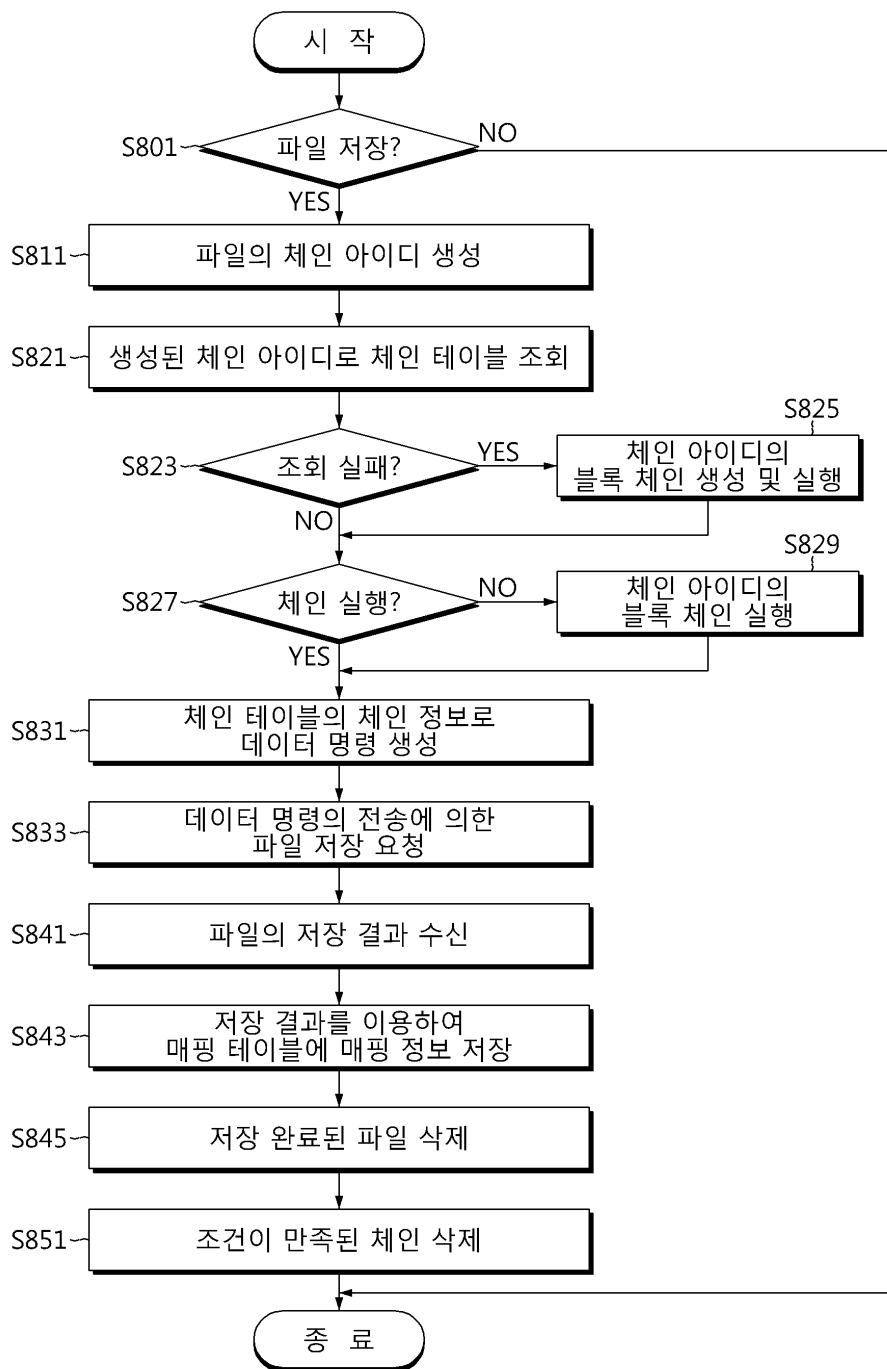
도면6



도면7



도면8



도면9

