이더리움 기반 공공정보 소프트웨어 사업산출물 관리 시스템 설계

이은주*, 김진욱*
*한국방송통신대학교 정보과학과
brent@knou.ac.kr, gnugi@knou.ac.kr

Ethereum-based deliverables management system design for public information software project

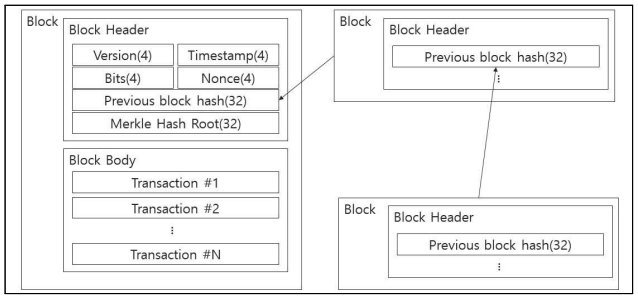
Eun Ju Lee*, Jin Wook Kim*
*Dept. of Computer Science, Korea National Open University

요 약

공공정보 소프트웨어(SW) 사업에서 프로젝트관리 방법론의 요구 산출물과 개발 산출물이 일치하지 않아 산출물의 누락이 발생하며, 대금 지급을 위한 별도의 프로세스를 진행해야 한다. 본 논문에서는 이더리움의 스마트 콘트랙트를 활용하여 이더리움 기반 공공정보 소프트웨어 사업산출물 관리 시스템을 설계하였다. 발주사의 요청에 따라 수행사가 등록한 산출물을 해시값으로 관리하여 산출물의 누락을 최소화하고, 요청된 모든 산출물이 승인되면 자동으로 수행사에 대금을 지급할 수 있도록 시스템을 설계하였다.

1. 서론

2008년 Satoshi Nakamoto가 제안한 블록체인은 P2P 기반의 분산 장부로 (그림 1)처럼 이전에 생성된 블록의 해시값을 새 블록의 헤더에 포함하여 블록을 체인으로 연결하였다.[1]



(그림 1) 블록체인의 구성

블록체인은 전자화폐로 사용될 뿐만 아니라 핀테크, 물류, 의료, 공공 분야 등에 대한 적용도 연구되고 있다.[2] 핀테크 분야에서는 암호화폐, 결재, 기부에 사용되고 있으며, 물류 분야에서는 공급망 사슬 전 과정에 적용되어 사용되고 있다. 의료 분야에서는 보험금 지급 정산 등에 활용을 준비하고 있으며, 공공 분야에서는 각종 민원 발급 및 전자투표에 블록체인 기술을 도입하고 있다.

공공정보 소프트웨어(SW) 개발사업을 진행할 시, 프로젝트관리 방법론에서 요구하는 산출물과 개발 과정에서 생성되는 산출물이 일치하지 않아 사업 완료 후 시스템 운영에 필요한 산출물이 누락되는 문제가 발생한다.[3] 발주사는 사업 완료 시 제출된 산출물의 누락 여부를 확인[4]해야 하며, 대금을 지급하기 위해 공문과 내부 기안을 통하는 등 별도의 프로세스가 필요하다.

산출물을 관리하기 위한 연구들이 진행되었다. [5]에서는 CVS 기반의 형상 관리 시스템을 설계/개발하여 산출물 요청 및 버전을 관리하였고, [6]에서는 SVN을 사용하여 저장소별 권한을 부여하는 방식으로 형상 관리 시스템을 구축하여 산출물을 관리하였다.

본 논문에서는 이더리움 기반 공공정보 소프트웨어 사업산출물 관리 시스템을 제안한다. 이 시스템은 이더리움의 스마트 콘트랙트 기술을 활용하였다. 사업산출물의 정보는 산출물 해시값을 이용하여 관리하고, 요청된 산출물이 모두 승인되면 대금을 지급하도록 설계한다.

2. 관련연구

2.1 공공정보 SW 사업 단계별 발주 프로세스

소프트웨어 진흥법[7] 제44조 SW사업의 과업 범위에 따르면 단계별 발주를 권고하고 있다. 단계별

발주는 요구사항 명확화를 위해 설계와 구현의 2단계로 분리하여 발주하는 방식이다. 설계 단계는 기획-요구사항분석-기본설계의 프로세스로 진행되며, 구현 단계는 기획-상세설계-구현-테스트-인수의 프로세스로 진행된다. 단계별 프로세스의 주요 산출물은 <표 1>과 같다.

<표 1> SW 단계별 발주 프로세스[8]

단계	프로세스	주요 산출물 명
설계	기획	정보화 추진 계획서, 사업 규모 산정서, 입찰 준비 서류
	요구사항 분석	요구사항 명세서, Use Case 시나리오, 사용자 검증/확인
	기본설계	기본설계서
구현	기획	정보화 추진 계획서, 상세 RFP, 입찰 준비 서류
	상세설계	DB 설계서, Test 명세서
	구현	단위 모듈설계서, Code, Test 시나리오
	테스트	Test 결과서, 품질검토서, Code 품질 결과서
	인수	설치 결과서, 인수 계획서, 시범운영 계획서

SW 단계별 발주는 설계와 구현 단계가 있으며, 각 단계의 마지막은 사업종료 프로세스를 수행해야 한다. [9]에 따르면 사업종료 프로세스는 사업 완료 검사, 인수 및 하자보수, 사업종료, 운영 및 유지보수의 프로세스로 진행된다. 사업 완료 검사에서 발주사는 산출물 검사를 통지받은 날로부터 14일 이내에 완료해야 한다. 사업종료에서 수행사가 관련 서류를 준비하여 대가지급을 청구하면 발주사는 청구받은 날로부터 근무일 기준 5일 이내로 대가를 지급해야 한다.

2.2 이더리움의 스마트 콘트랙트

비트코인은 암호화폐 기능만 제공하는 반면 이더리움은 화폐뿐만 아니라 계약서, 전자투표 등 다양한 시스템을 개발할 수 있는 플랫폼을 제공한다. 이더리움 기반으로 구축된 탈중앙화 애플리케이션을 디앱(DApp)이라 부르며 Solidity 프로그램을 사용하여 스마트 콘트랙트를 개발한다.[10] (그림 2)는 변수를 정수로 입력받는 Solidity 프로그램의 예시이다.

```
pragma solidity >=0.7.0;  // 버전

contract SmartContract{  //콘트랙트명
  uint data;  // 정수형 변수
  // 변수를 입력
  function setData(uint _data) public{
      data = _data;
  }
  // 변수를 리턴
  function getData() public view returns(uint _data){
      _data = data;
  }
}
```

(그림 2) 스마트 콘트랙트 소스 예시

스마트 콘트랙트는 사용자 간 합의된 계약서로 이더리움 블록의 트랜잭션에 함수와 데이터 집합으로 저장된다. 계약자 간의 계약정보를 스마트 콘트랙트로 정의하여 계약 조건이 충족되면 다음 프로세스를 실행하는 등의 신뢰성 있는 시스템 개발이 가능하다.[11]

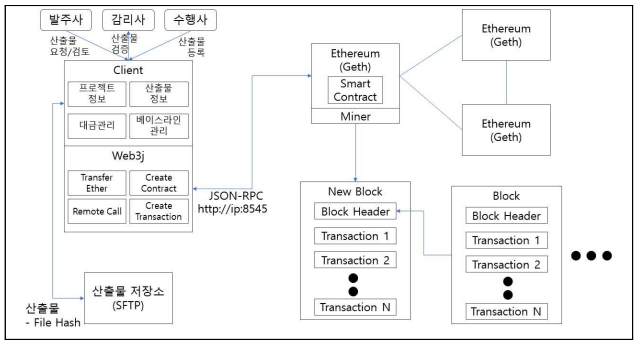
이더리움의 스마트 콘트랙트를 사용하여 전자화폐를 전송하거나 전자문서 정보를 블록에 저장하는 등에 활용이 많아지고 있다. Król 등[12]은 스마트 콘트랙트에서 아웃소싱 업체가 개발한 프로그램이 테스트 소스 코드를 통과하면 보상이 지급되는 시스템으로 ChainSoft를 제안하였다. ChainSoft에서는 GitHub로 소스 형상을 관리하고 Travis CI를 통해 소스 코드의 검증을 자동화하였다. Nizamuddin 등[13]은 디지털 문서의 버전을 관리하기 위해 전자문서 정보를 스마트 콘트랙트에 등록하고, 전자문서는 IPFS에 저장하였다. 그리고 승인자의 2/3 이상이 승인하면 신규 버전으로 등록되는 시스템을 제안하였다.

소프트웨어 형상 관리와 대금지불 그리고 전자문서의 버전관리에 관한 연구가 이루어지고 있으나, 해당 연구는 소스 코드의 정확한 동작에 따른 검증을 대금 지급 기준으로 보고 있다. 이에 산출물 누락을 최소화하고 산출물이 모두 제출되었을 때 자동으로 대금을 지급할 수 있는 산출물 관리 시스템이 필요하다.

3. 이더리움을 활용한 산출물 관리 시스템 설계

본 논문에서는 이더리움 기반 공공정보 소프트웨어 사업산출물 관리 시스템을 설계한다. 구체적으로, 발주사가 산출물을 요청/검토하고, 수행사는 산출물

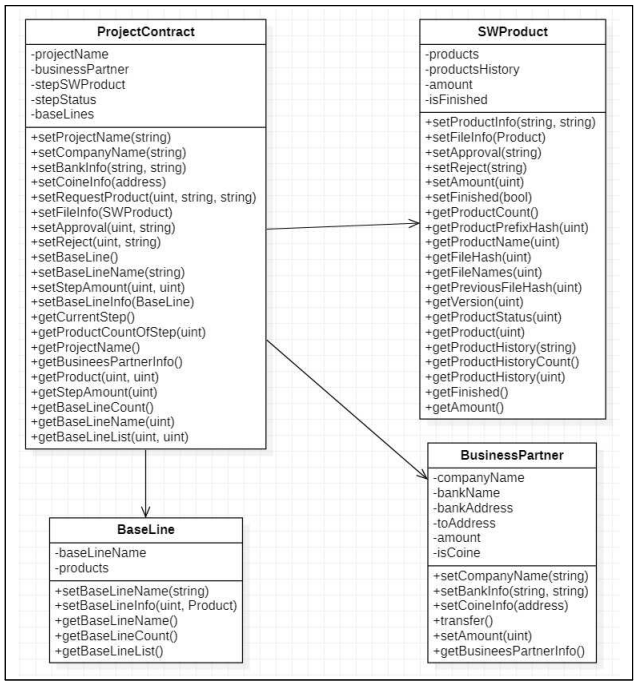
을 등록하며, 감리사는 산출물을 검증하고, 발주사에서 모든 산출물을 승인하면 대금이 자동으로 수행사에 지급되는 시스템을 설계한다. 시스템은 공공기관(발주사)에서 프라이빗 블록체인으로 구성하고 운영하는 것으로 가정한다.



(그림 3) 제안시스템의 구성도

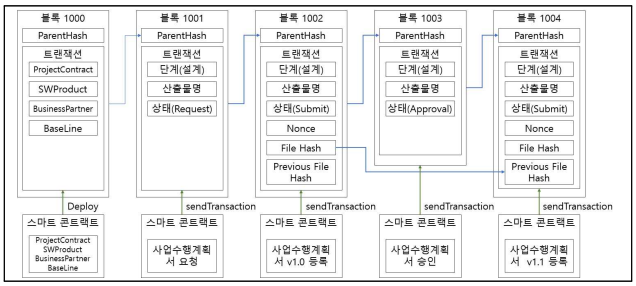
시스템은 (그림 3)과 같이 사용자의 입력을 받는 Client와 블록체인을 구성할 이더리움(Geth), 산출물 저장소(SFTP), 그리고 Client와 이더리움 간의 통신을 담당하는 Web3j로 구성된다.

사용된 스마트 콘트랙트는 (그림 4)와 같이 4개로 구성된다. ProjectContract에서 프로젝트를 관리하고, SWProduct에서 산출물 정보를 관리하며 BusinessPartner는 수행사 정보와 계약정보를 관리한다. BaseLine은 산출물의 최종본을 관리하는 스마트 콘트랙트이다.



(그림 4) 스마트 콘트랙트의 클래스 다이어그램

스마트 콘트랙트를 사용하여 블록체인에 저장되는 정보는 (그림 5)와 같다. 스마트 콘트랙트를 배포(Deploy)하면 블록체인의 트랜잭션에 계약정보가 저장된다. 산출물의 요청/등록/승인 작업 시 배포된 스마트 콘트랙트가 호출(sendTransaction)되고 입력된 정보는 블록체인의 트랜잭션에 정보가 저장된다.



(그림 5) 사업산출물 관리 시스템 블록 구성

산출물은 개발 과정에서 지속적인 변경이 발생하므로 버전 정보를 포함하는 산출물 해시값을 계산하여 산출물의 변경 추적성을 관리하고자 한다. 수행사에서 산출물을 등록하면 산출물의 해시값을 계산하여 파일명으로 사용하고, 블록체인의 트랜잭션에 해시값을 저장한다. 트랜잭션에 저장된 해시값은 저장소에서 산출물을 불러올 때 사용한다. 산출물의 해시값 계산은 산출물 파일(File), 이전 산출물 파일의 해시값(Previous File Hash), Nonce를 이용하여 계산하며 해당 산출물에 맞는 bits로 시작하는 해시값을 구할 때까지 반복한다. (그림 6)은 산출물 파일 해시값을 구하는 알고리즘이다.

```
Input step, type, version, previousFileHash, file
Initialize prefix = step + type + version
Initialize fileHash="", nonce=0
while HASH(prefix) != substring(fileHash,0,5) do
    fileHash=HASH(file,previousFileHash,Nonce)
    nonce++;
```

(그림 6) 산출물 파일 해시값을 구하는 알고리즘

해당 산출물에 맞는 bits는 산출물 ID로부터 계산한다. 산출물 ID는 6자리 10진수로, 단계별 프로세스(step) 1자리, 산출물 종류(type) 2자리, 버전(version) 3자리(Major 2자리, Minor 1자리)로 구성한다. 이때, 단계별 프로세스는 설계의 요구사항 분석 1번, 기본설계 2번, 구현의 상세설계 3번, 구현 4번, 테스트 5번, 인수 6번으로 정의한다. 산출물 종류는 요구사항명세서 01, Use Case 시나리오 02등으로 정의한다. 그리고 버전은 초기 작성된 산출물의 경우 Major 01, Minor 0으로 하여 010으로 정의한다.

6자리 10진수 산출물 ID를 16진수로 변환하면 5자리(20 bits)로 변환된다. 변환된 5자리로 시작하는

해시값을 찾아 스마트 콘트랙트에 산출물의 정보를 저장한다. 예를 들면, 설계 단계의 요구사항 분석 프로세스에서 만들어지는 요구사항명세서 2.1버전의 산출물 ID는 101021로 정의되며, 해시값은 18A9D로 시작된다.

발주사가 요청한 산출물이 모두 등록되고 승인되면 (그림 7)과 같이 대금을 자동으로 지급한다. 이 알고리즘은 발주사가 산출물을 승인할 때마다 수행되며, 발주사가 요청한 모든 산출물이 승인되었는지 확인한다. 모두 승인되었다면 수행사의 이더리움 주소에 대금을 송금한다.

```
Input toAddress, amount
Input RequestDelivers[], ApprovalDelivers[]
Initialize isFinish=true; isCheck=false
If RequestDelivers.length not equal ApprovalDelivers.length:
 isFinish = false
For RequestDelivers length do
 isCheck=false
 For ApprovalDelivers length do
  If RequestDelivers.name equal ApprovalDelivers.name:
   isCheck=true
 If isCheck equal false:
  isFinish = false
If isFinish equal true:
 toAddress.transfer(amount)
```

(그림 7) 산출물 승인 후 대금을 지급하는 알고리즘

4. 결론

본 논문에서는 이더리움 기반 공공정보 소프트웨어 사업산출물 관리 시스템을 설계하였다. 이 시스템은 이더리움의 스마트 콘트랙트를 사용하여 발주사와 수행사 간에 제출할 산출물 정보를 블록체인에 저장한다. 수행사는 요청된 산출물을 등록하고, 발주사는 확인 후 승인한다. 모든 산출물이 승인되면 수행사의 이더리움 주소로 대금을 지급되게 된다.

수행사는 요청된 산출물을 등록하고 검수가 완료되어야 대금이 지급되므로 산출물을 등록해야 한다. 발주사는 수행사가 제출한 산출물을 등록된 순서로 승인하여 점검 시간의 확보가 가능하다. 산출물이 모두 승인되면 자동으로 대금이 지급되어 내부 프로세스가 간단해지며 산출물의 누락 여부 확인이 쉽다. 산출물의 정보가 블록체인에 등록되어 산출물의 이력 관리가 투명해지고 최종 버전의 산출물 확인이 쉬워진다. 산출물을 해시값으로 관리하여 식별 및 분류를 할 수 있으며, 산출물 해시값이 순차적인 증가로 버전관리가 쉬워진다.

향후 연구로 설계한 시스템의 구현과 public 이더리움으로 확장하여 실제로 대금을 지급할 수 있는 환경을 구축하고자 한다.

참고문헌

[1] Satoshi Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System”, 2008.

[2] 양영균, 천세학, “블록체인 기술의 활용 현황과 발전 방안에 관한 연구: 해외의 암호화폐 정책을 중심으로”, 상업교육연구, vol.33, no.2, pp. 47-70, 2019.

[3] 김의정, 김희천, “발주자 관점의 IT 프로젝트 산출물 연구 - 행정기관의 소규모 프로젝트를 중심으로”, 한국정보처리학회, Vol.21 No.2, pp. 605-608, 2014.

[4] “전자정부지원사업 정보화전략계획(ISP) 산출물 점검 가이드”, 행정자치부·한국정보화진흥원, 2015.

[5] 이정훈, “CVS 기반의 소프트웨어 형상관리 시스템”, 석사학위논문, 충남대학교, 2006.

[6] 도성룡, “소규모 조직을 위한 CMMI 기반의 형상관리 프로세스 구축” 석사학위논문, 상명대학교, 2009.

[7] “소프트웨어진흥법”, 과학기술정보통신부, 법률 제17348호, 2020

[8] “소프트웨어 단계별 발주 가이드”, NIPA, 2019.

[9] “정보화사업 단계별 관리·점검 가이드 V3.0”, 미래창조과학부·행정자치부·한국정보화진흥원, 2015.

[10] Ethereum https://ethereum.org/ (accessed September 30, 2021)

[11] Solidity Programming Language https://soliditylang.org/ (accessed September 30, 2021)

[12] M. Król, S. Reñé, O. Ascigil and I. Psaras, "ChainSoft: Collaborative software development using smart contracts", Proc. 1st Workshop Cryptocurrencies Blockchains Distrib. Syst. (CryBlock), pp. 1-6, 2018.

[13] N. Nizamuddin, K. Salah, M. A. Azad, J. Arshad and M. H. Rehman, "Decentralized document version control using ethereum blockchain and IPFS", Comput. Electr. Eng., vol. 76, pp. 187-197, Jun. 2019.