

<https://doi.org/10.7236/IIBC.2020.20.4.115>  
IIBC 2020-4-16

## Blockchain Watchdog: 분산 저장소와 스마트 컨트랙트 코드 연동 실시간 블록체인 관제시스템

### Blockchain Watchdog: Real-time Blockchain Surveillance System Connecting Smart Contract Code and Distributed Storage

홍준기\*, 김순태\*\*, 류덕산\*\*

Joongi Hong\*, Suntae Kim\*\*, Duksan Ryu\*\*

**요약** 블록체인의 참여가 증가하면서 생태계 유지를 위한 거버넌스가 활성화되고 있다. 거버넌스는 블록체인을 유지하고 발전시키기 위해 관리/감독이 이루어져야 한다. 하지만 블록체인의 사용량이 증가함에 따라 사람이 일일이 관리하기엔 한계가 존재한다. 또한, 이더리움부터 적용된 스마트 컨트랙트라는 새로운 거래 형태가 가능함에 따라 상태 값에 따라 결정되는 거래에 대해 검증이 필요하다. 본 연구는 거버넌스의 블록체인 생태계 유지와 발전을 위한 도구로 블록체인 스마트 컨트랙트의 상태 모니터링을 기반으로 하는 관제시스템(Blockchain Watchdog)을 제안하며, 관제시스템의 구현 및 실험을 진행하였다. 사례 연구로 Ropsten과 Ganache를 활용한 사설 네트워크에서 실험을 진행하였으며, 실시간 정보수집과 대시보드와 그래프, 차트를 활용한 시각화를 볼 수 있었다. 우리는 Blockchain Watchdog을 통해 블록체인 거버넌스의 생태계 유지와 발전에 기여할 것이라 기대한다.

**Abstract** As the participation of blockchain increases, governance to maintain the ecosystem is being activated. Governance must be managed/supervised to maintain and improve the blockchain. However, as the usage of blockchain increases, there is a limit for people to manage it individually. In addition, smart contracts applied from Ethereum need to be verified for trade that are determined according to status values, as new trade types are possible. This research proposes a surveillance system, Blockchain Watchdog, based on the state monitoring of the blockchain smart contract as a tool for maintaining and developing the blockchain ecosystem of governance, and implemented and tested the surveillance system. As a case study, we conducted experiments on Ropsten and Ganache, and were able to see real-time information collection and visualization using dashboards, graphs, and charts. We expect to contribute to the maintenance and development of the ecosystem of blockchain governance through Blockchain Watchdog.

**Key Words** : Blockchain, Smart Contract, Smart Contract State, Surveillance System, Ethereum

\*준회원, 전북대학교, 소프트웨어공학과  
\*\*정회원, 전북대학교, 소프트웨어공학과  
접수일자 2020년 6월 15일, 수정완료 2020년 7월 15일  
게재확정일자 2020년 8월 7일

Received: 15 June, 2020 / Revised: 15 July, 2020 /  
Accepted: 7 August, 2020

\*\*Corresponding Author: stkim@jbnu.ac.kr  
Department of Software Engineering, ChonBuk National University, Korea.

## I. 서 론

2009년 비트코인(Bitcoin)과 함께 등장한 블록체인 기술은 이더리움(Ethereum)과 하이퍼레저(Hyperledger), 이오스(EOS) 등 다양한 블록체인의 형태로 발전하며 최근 사람들에게 관심을 얻고 급격하게 성장하였다. 대표적으로 비트코인과 이더리움은 2020년 1분기 기준으로 각각 약 30만 개와 70만 개의 트랜잭션이 발행되어 처리되고 있다<sup>[1][2]</sup>. 이더리움<sup>[3]</sup>의 경우 스마트 컨트랙트<sup>[4]</sup>를 적용하여 단순 돈거래가 아닌 비즈니스에 따른 거래를 가능케 하여 새로운 패러다임을 제시하였으며, 비트코인 다음의 인기를 누리고 있다<sup>[5]</sup>. 블록체인의 관심과 참여가 증가함에 따라 블록체인을 활용하는 사용자들의 거버넌스가 활성화되고 있으며, 생태계의 유지와 발전을 위해 힘쓰고 있다. 이더리움에서 거버넌스의 이해관계자로는 크게 블록체인 자체의 개발을 위한 코어 개발자, 블록 채굴자, Dapp(Decentralized Application) 개발자와 일반 사용자 등으로 나뉜다<sup>[6]</sup>.

블록체인 생태계 유지를 위해 거버넌스는 블록체인에 대한 관리/감독을 통하여 문제 파악과 해결을 할 수 있어야 한다. 하지만 이해관계자들의 참여가 증가함에 따라 발행되는 트랜잭션의 양이 사람이 일일이 관리하기에는 방대한 양이기 때문에 한계가 존재한다. 또한, 이더리움의 경우 스마트 컨트랙트를 활용한 다양한 거래가 진행된다. 단순 코인 거래뿐만 아니라 비즈니스에 따른 처리도 가능하게 됨에 따라 사기, 비즈니스 로직의 취약점, 악의적인 사용 및 공격 등의 사례들이 나오고 있으며, 이를 예방하기 위한 관리/감독 방법이 필요하다. 대표적인 사건으로는 2016년 6월에 발생한 DAO 해킹 사건이 있다. DAO 해킹 사건은 해커가 생성한 계좌로 243만 이더 이상을 출금해 갔으며 당시 한화 가치로 약 750억 원이 해킹당했다<sup>[7]</sup>.

본 연구에서는 블록체인 생태계에서 발생하는 다양한 거래 정보를 스마트 컨트랙트 중심으로 정보를 수집하고 거버넌스가 문제 상황 파악 및 해결을 지원하는 Blockchain Watchdog(블록체인 관제시스템)을 제안한다. Blockchain Watchdog은 첫 번째로 스마트 컨트랙트의 상태 변화 이력에 대한 모니터링을 진행하고 대시보드, 차트, 그래프 등을 활용하여 상태 정보를 표현한다. 사용자들은 문제 상황 발생 시 문제 원인 추적이 가능하다. 두 번째로 대량의 정보를 자동으로 관리하여 사람이 관리하기에 한계가 있는 부분을 해소하고자 한다. 세 번째로 스마트 컨트랙트의 상태 값을 일반 사용자와

개발자가 알기 쉽게 파악할 수 있고 프로그램에 대한 확인 및 검증을 위한 도구로서 활용될 수 있게 하고자 한다.

블록체인에 관제시스템을 적용하기 위해서는 여러 도전과제가 존재하고 있다. 먼저 스마트 컨트랙트에 존재하는 상태가 어떤 것들이 있는지에 대한 파악과 상태 변수의 실제 값이 무엇인지 파악해야 하는 기술적인 문제가 존재한다. 두 번째로 실시간으로 발생하는 대량의 동적 데이터 처리와 성능적인 측면의 문제가 있다. 마지막으로 개발자 또는 일반 사용자들이 컨트랙트에 대한 정보를 쉽게 파악할 수 있도록 효과적인 시각화에 대한 방법이 필요했다.

우리는 Blockchain Watchdog을 연구 및 개발하기 위하여 스마트 컨트랙트를 활용하는 가장 대표적인 플랫폼인 이더리움을 대상으로 연구 및 개발을 진행하였다. 또한, 접근 방법을 직접 구현해보았고 실제 이더리움 테스트넷인 Ropsten과 Ganache<sup>[8]</sup>로 사례 연구를 진행하였다. 블록체인에 배포되어 실제로 동작하는 스마트 컨트랙트에 변화하는 상태 정보를 수집하고 그에 따른 대시보드, 차트, 그래프 등을 실시간으로 확인할 수 있었다.

## II. 배경지식 및 관련 연구

본 연구에서 제안하는 블록체인 스마트 컨트랙트 관제시스템에 대한 이해를 돕기 위해 블록체인과 스마트 컨트랙트, 관제시스템에 대한 이해, 현재 진행되고 있는 관련된 연구를 함께 소개한다.

1. 블록체인(Blockchain) P2P(Peer to Peer) 네트워크와 합의 알고리즘을 기반으로 한 탈중앙화 분산 저장 시스템이다. 다수의 노드가 참여하여 네트워크를 형성하며, 합의 과정과 해시의 특성을 활용하여 데이터의 위변조를 방지할 수 있는 기술이다. 공개(Public)/허가 또는 폐쇄형(Permissioned or Private) 블록체인의 형태로 존재한다<sup>[9]</sup>.

2. 스마트 컨트랙트(Smart Contract)와 상태(State) 스마트 컨트랙트는 닉사보(Nick Szabo)에 의해 정립된 개념이며, 프로그래밍 언어로 된 계약서이다. 이더리움에서 블록체인에 처음으로 도입되었으며 솔리디티(Solidity)<sup>[10]</sup> 언어를 활용하여 작성할 수 있다. 블록체인에 배포되어 사용되며 위변조할 수 없는 계약서를 실현하였고 계약 조건 충족 시 프로그램에 의한 계약 강제 이행이라는 장점이 있다. 그림 1은 솔리디티를 활용하여 작성한 간단한 경매 컨트랙트 예제이다.

```
1 pragma solidity >=0.4.22 <0.7.0;
2
3 contract SimpleAuction {
4
5     address payable public beneficiary;
6     uint public auctionEndTime;
7
8     address public highestBidder;
9     uint public highestBid;
10
11     mapping(address => uint) pendingReturns;
12     bool ended;
13
14     event HighestBidIncreased(address bidder, uint amount);
15     event AuctionEnded(address winner, uint amount);
16
17     constructor(
18         uint _biddingTime,
19         address payable _beneficiary
20     ) public {
21         beneficiary = _beneficiary;
22         auctionEndTime = now + _biddingTime;
23     }
24
25     function bid() public payable {
26         require(
27             now <= auctionEndTime,
28             "Auction already ended."
29         );
30     }
31 }
```

그림 1. Simple Auction - 스마트 컨트랙트 예제  
Fig. 1. Simple Auction - Smart Contract Example

상태란 계약을 위해 사용되는 변화하는 값으로 값에 따라 계약 이행 여부가 결정된다. 솔리디티 코드에서는 전역변수처럼 선언된 변수들을 의미한다. 예제의 상태로는 beneficiary, auctionEndTime, highestBidder, highestBid, pendingReturns, ended 등이 해당된다.

3. 관제시스템 관리자가 해당 도메인의 정보를 수집하고 발생할 수 있는 문제에 대비하고 해결하기 위해 사용하는 관리와 통제를 위한 시스템이다. 관제는 많은 분야에서 관리를 위한 일반적인 방법으로 쓰이고 있으며, 자동차와 비행기, 배 등의 교통, 보안을 위한 CCTV 영상, 민감한 데이터, 물류 등의 관리를 위해 사용되고 있다. 소프트웨어 분야에서는 대표적으로 클라우드에 적용되고 있으며, 클라우드에서 동작하는 애플리케이션 정보를 모니터링하여 프로파일링, 최적화 작업 등을 수행하고 하드웨어 요구사항에 유연하게 대응하여 소프트웨어의 서비스 품질을 높이는 데 사용된다<sup>[11][12]</sup>.

4. SmartInspect, SmartAnvil<sup>[13][14]</sup> SmartAnvil은 SmartInspect의 확장된 연구이며, SmartInspect은 이더리움 스마트 컨트랙트의 상태를 솔리디티 코드의 변수 값과 매핑하여 실제 값을 추출하는 연구를 진행하였다. 본 연구의 기반 기술로 컨트랙트의 소스 코드와 주소를 활용하여 이더리움 네트워크에 배포된 상태 정보를 추출하는 데 활용하였다. SmartInspect에서 확장된 SmartAnvil의 경우 본 연구와 달리 컨트랙트 수집한 상태값을 활용하여 정적 분석 도구로써 활용하여 동적인 데이터를 실시간으로 제공하지 않는다.

5. 이더스캔(Etherscan) 이더스캔은 이더리움 블록체

인의 블록, 트랜잭션, 계정 등 블록체인의 전반적인 데이터 정보를 제공하는 모니터링 도구이다. 하지만 스마트 컨트랙트에 대한 전체적인 정보는 제공하지만, 변화하는 상태 값에 대한 정보를 제공하지는 않는다.

6. 리믹스(Remix) 리믹스<sup>[15]</sup>는 이더리움의 스마트 컨트랙트 개발을 위한 Web 기반의 IDE(Integrated development environment)이다. 소스 코드 작성뿐만 아니라 배포, 디버깅, 컨트랙트 호출도 가능하다. 리믹스는 디버깅 모드에서 트랜잭션의 상태 변화를 확인할 수 있는 기능을 제공하고 있다. 하지만 한 트랜잭션에 의해 변화한 상태만을 확인할 수 있어 전체적인 흐름과 여러 컨트랙트를 한꺼번에 관리하는 데는 무리가 있다.

### III. Blockchain Watchdog

Blockchain Watchdog은 그림 2와 같이 총 4개의 계층 구조로 구성되어 있다. 스마트 컨트랙트를 분석하고 관리하기 위한 Parser 계층, 이더리움 블록체인과 연동하여 실시간으로 컨트랙트의 정보를 관찰하는 Observer 계층, 컨트랙트의 정보를 기반으로 하여 발생할 수 있는 문제를 검사하는 Analyzer 계층이 있다. 아래에서는 계층별 상세 내용을 다룬다.

1. Parser Layer 스마트 컨트랙트 소스 코드 분석을 위한 계층이다. Source Code Parser와 Smart Contract Manager 2개의 모듈로 구성된다.

Source Code Parser에서는 사용자로부터 솔리디티 소스 코드를 입력받아 AST를 활용한 토큰화 작업을 진행한다. 토큰화 작업이란 Antlr<sup>[16]</sup>를 활용하여 소스 코드의 AST를 추출하고 컨트랙트의 상태 변수를 찾아내는 작업이다. 토큰화 작업 후에 Smart Contract Manager에 의해 컨트랙트는 저장소에 저장 및 관리 된다. 토큰화된 데이터는 상태 변수명 활용과 데이터 변환을 위해 활용된다. 그림 3와 같이 JSON의 형태로 상태 변수의 데이터 타입과 변수명, 크기 정보가 담긴다.

2. Observer Layer 이더리움 네트워크와 연동하여 배포된 스마트 컨트랙트의 상태 정보수집 계층이다. Blockchain Monitoring Service, Data Converter, Blockchain Communicator 3개의 모듈로 구성된다.

Blockchain Monitoring Service는 블록 생성에 따른 실시간 컨트랙트 상태 정보수집을 총괄하는 모듈이다. Data Converter 및 Smart Contract Manager, Blockchain Communicator와 상호작용한다. 수집된

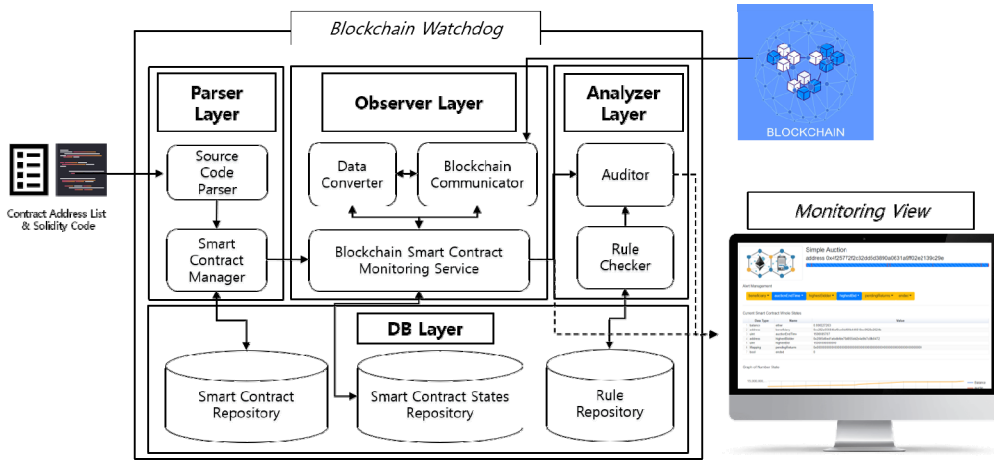


그림 2. Blockchain Watchdog(블록체인 관제 시스템) 아키텍처  
Fig. 2. Blockchain Watchdog(Blockchain Surveillance System) Architecture

정보는 JSON 형태로 Smart Contract State Repository에 저장 후 이상 여부 분석을 위해 Auditor에 전달된다. Blockchain Communicator는 Web3<sup>[17]</sup>를 활용하여 블록체인으로부터 정보를 가져오는 역할을 담당한다. 블록 생성 시마다 블록 정보를 가져오며, Smart Contract Manager에 등록된 컨트랙트 주소 테이블을 활용하여 Blockchain Monitoring Service에서 필터링한다.

```

.....
{"type": "address",
 "name": "highestBidder",
 "size": 160 },
{"type": "uint",
 "name": "highestBid",
 "size": 256},
.....

```

그림 3. 상태 Token JSON  
Fig. 3. State Token JSON

Data Converter에서는 토큰화된 상태 변수 정보와 블록체인 네트워크에 인코딩된 상태 정보를 가져와 데이터 매핑 및 변환 작업을 처리한다. 해당 모듈은 Smart Inspect에서 제안한 방법을 기반으로 하여 동작한다.

3. Analyzer Layer 사용자가 설정한 한계 범위 또는 값을 기반으로 하여 수집되는 스마트 컨트랙트의 상태 정보에 대한 경고 및 알람을 위한 계층이다. Auditor와 Rule Checker 2개의 모듈로 구성된다.

Auditor는 블록체인 네트워크로부터 얻어온 데이터를 분석한다. Rule Checker와 상호작용하며, Rule을 위반하는 상황이 발생하면 모니터링 화면에 경고 표시와 알람(이메일 또는 SMS 등)을 전송한다.

Rule Checker는 Rule Repository에 저장된 Rule을 기반으로 하여 데이터의 이상 여부를 판단하게 된다. Rule에는 숫자 상태일 경우 범위를 설정하는 것과 특정 값이 들어오면 경고 또는 알람을 받을 수 있게 설정할 수 있다. 문자열 데이터의 경우 특정 값만이 입력되거나 입력되지 못하도록 하는 방식으로 설정할 수 있다.

4. DB Layer 스마트 컨트랙트 정보와 규칙을 저장하기 위한 저장소 계층이다. Smart Contract, Smart Contract State, Rule 저장소 3개가 존재한다. Smart Contract 저장소에는 컨트랙트의 주소와 소스 코드, 토큰화된 상태 정보가 담겨 있다. Smart Contract State 저장소에는 등록된 컨트랙트 별 상태 정보의 이력이 저장된다. 마지막으로 Rule 저장소에는 사용자로부터 입력 받은 규칙 정보가 담겨 있다. 실시간 대용량 처리를 위해 MongoDB를 선택하여 진행하였다.

#### IV. 사례 연구

본 연구에서는 이더리움 네트워크에 배포된 스마트 컨트랙트의 모니터링 가능 여부와 실시간성, 데이터 정확성 등을 판별해보기 위하여 Blockchain Watchdog을 Ganache와 Ropsten 대상으로 실험하였다.

실험을 위하여 그림 4와 같이 웹 애플리케이션 형태로 구현하였으며, 구현에 사용된 하드웨어와 소프트웨어 환경은 OS: Windows10, CPU: AMD RYZEN 1700, RAM: Samsung 24GB, Ganache:2.4.0, Web3:1.2.7, Express:4.16.1, MongoDB:4.2.1로 구성되었다.

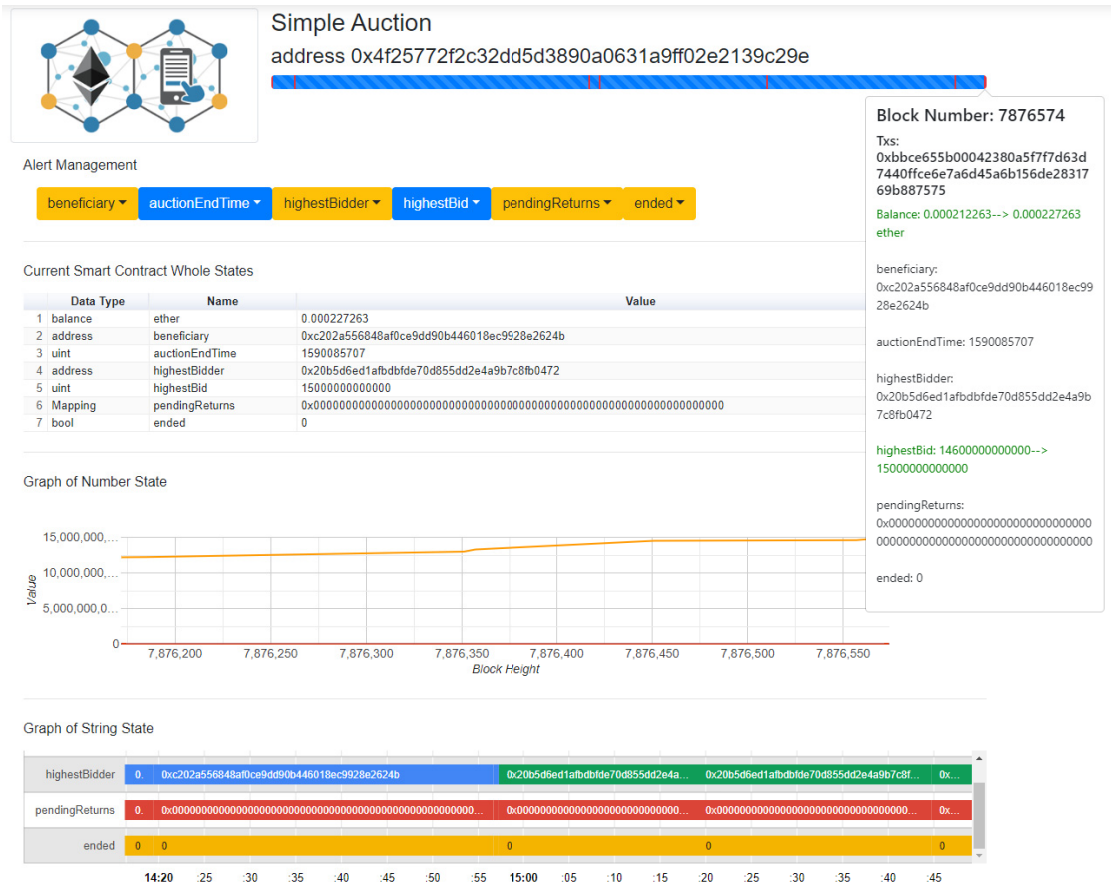


그림 4. 이더리움에 적용한 Blockchain Watchdog  
 Fig. 4. Blockchain Watchdog Applied to Ethereum

아래에서는 우리가 앞에서 언급한 3가지 도전과제에 대해 분석해 볼 것이다. 실험 전에 미리 스마트 컨트랙트는 네트워크에 배포되었으며, 그림 1 예제를 사용하였다.

RQ1) 스마트 컨트랙트 상태 파악과 실제값 추출 가능 여부: 예제에는 상태 변수는 총 6개이며, 그림 4의 중간에 있는 표에서 확인할 수 있듯이 모든 상태값들이 표현되는 것을 확인할 수 있다. 추가적으로 이더값도 나타난다. 예제에서는 빠져있지만 String, int, Array (size) 관련 정보도 확인 가능했다.

RQ2) 관제시스템의 실시간 대량 데이터 모니터링 가능 여부: Ganache의 경우 하나의 트랜잭션 발생시 하나의 블록이 생성되기 때문에 지연 시간 및 동기화 시간을 필요로하지 않아 즉각적인 데이터 처리가 가능하였다. 하지만 Ropsten의 경우 블록화 및 동기화의 지연시간이 존재하여 실시간 처리는 가능하지만 상태 변화의 누락이 발생하는 것을 확인하였다. 구체적으로 블록이 생성되고

헤더는 공유되었지만 블록의 전체 데이터는 동기화가 되지 않아 데이터의 누락이 발생한다. 대량의 데이터 처리의 경우 하나의 블록이 발생할 경우 처리 시간이 약 5초 정도였으며, 병렬처리를 통해 대량의 데이터가 문제 없이 처리되는 것을 볼 수 있었다.

RQ3) 효과적인 스마트 컨트랙트 상태 시각화 적용: 우리는 총 5가지의 요소를 제공한다. 첫 번째에서는 블록 순으로 상태 변화 타임라인을 제공한다. 상태 변화시 붉은 줄로 표시되며 해당 지점에서 변화된 정보를 조회할 수 있다. 그 아래 경고 관리에서는 상태별 경고 설정 기능을 제공한다. 중간에 표에서는 현 시점의 스마트 컨트랙트 상태 전체 정보를 제공하고, 아래의 그래프에서는 숫자형 데이터와 문자형 데이터를 구분하여 표현한다. 숫자형의 경우 값의 변화에 대한 그래프를 나타내며, 문자형 데이터의 경우 값의 변화 시점을 표현하는 타임라인의 형태로 시각화되었다.

## V. 결 론

본 연구는 블록체인 생태계 거버넌스의 관리/감독을 지원하고자 Blockchain Watchdog(블록체인 관제시스템)을 제안하였다. 또한, 실제로 시스템을 일부 개발 및 구현하여 사설 네트워크 기반의 Ganache와 Ropsten 테스트넷에 적용하여 실험을 진행하였다. 실험 결과 블록체인에서 배포된 컨트랙트의 상태 정보를 추출하여 대시보드와 그래프, 차트 등의 형태로 데이터를 시각화하여 표현하였지만 Ropsten의 경우 블록 정보 누락 문제가 존재하는 것을 확인했다. 상태 모니터링이 가능하게 됨에 따라 우리는 본 연구를 통해 블록체인 거버넌스의 생태계 유지 및 발전에 기여할 수 있을 것이라 기대한다.

향후 연구로는 구현하지 못한 경고 및 알림을 추가하고 이오스, 트론 등 다른 스마트 컨트랙트를 활용하는 블록체인으로 확장하여 적용하는 연구를 진행할 예정이다.

## References

- [1] (Apr., 2020) Blockchain.com. [Online]. Available: <https://www.blockchain.com/ko/charts>
- [2] (Apr., 2020) Etherscan [Online]. Available: <https://etherscan.io/>
- [3] Buterin, Vitalik. "Ethereum white paper, 2014." URL <https://github.com/ethereum/wiki/wiki/White-Paper> (2013).
- [4] N. Szabo, "Smart Contracts: Building Blocks for Digital Free Markets", *Extropy Journal of Transhuman Thought*, 1996, DOI: 10.1200/JCO.2011.40.6546.
- [5] JangWon Choi "A Review of the Legal Nature that Users of the Virtual Currency Exchange Obtain and the Compensation Responsibility for the Damages Caused By Internet Problems or Network Errors" *Journal of the Korea Academia-Industrial cooperation Society v.19, no.11, 2018*, pp. 287-294 DOI: 10.5762/KAIS.2018.19.11.287
- [6] (May., 2020) ethhub. [Online]. Available: <https://docs.ethhub.io/ethereum-basics/governance>
- [7] Mehar, Muhammad Izhar, et al. "Understanding a revolutionary and flawed grand experiment in blockchain: the DAO attack." *Journal of Cases on Information Technology (JCIT) Vol. 21 No. 1, 2019*, pp. 19-32. DOI: 10.4018/JCIT.2019010102
- [8] (Apr., 2020) Ganache. [Online]. Available: <https://www.trufflesuite.com/ganache>
- [9] Zheng, Zibin, et al. "Blockchain challenges and opportunities: A survey." *International Journal of Web and Grid Services (IJWGS)*, Vol. 14, No. 4, 2018 pp. 352-375. DOI: 10.1504/IJWGS.2018.095647
- [10] (May., 2020) Solidity document. [Online]. Available: <http://solidity.readthedocs.io>
- [11] Shao, Jin, et al. "A runtime model based monitoring approach for cloud." *2010 IEEE 3rd International Conference on Cloud Computing*. IEEE, 2010. DOI: 10.1109/CLOUD.2010.31
- [12] J. Shao, H. Wei, Q. Wang and H. Mei, "A Runtime Model Based Monitoring Approach for Cloud," *2010 IEEE 3rd International Conference on Cloud Computing*, Miami, FL, 2010, pp. 313-320, DOI: 10.1109/CLOUD.2010.31.
- [13] S. Bragagnolo, H. Rocha, M. Denker and S. Ducasse, "SmartInspect: solidity smart contract inspector," *2018 International Workshop on Blockchain Oriented Software Engineering (IWBOSE)*, Campobasso, 2018, pp. 9-18, DOI: 10.1109/IWBOSE.2018.8327566.
- [14] Ducasse, Stéphane, et al. "Open-Source Tool Suite for Smart Contract Analysis." *Blockchain and Web 3.0: Social, Economic, and Technological Challenges* (2019).
- [15] (Apr., 2020) Remix. [Online]. Available: <http://remix.ethereum.org/>
- [16] (Jul., 2019) Antlr [Online]. Available: <https://wwwantlr.org/>
- [17] (Mar., 2020) Web3.js [Online]. Available: <https://web3js.readthedocs.io/>

## 저 자 소 개

### 홍 준 기 (준회원)



- 2019.2~: 전북대학교 석사 재학
- 2019: 전북대학교 학사
- Email: rlwns012@jbnu.ac.kr

김 순 태(정회원)



- 2014~: 전북대학교 소프트웨어공학과  
부교수
- 2010: 서강대학교 공학박사
- 2007: 서강대학교 공학석사
- Email: stkim@jbnu.ac.kr

류 덕 산 (정회원)



- 2018~: 전북대학교 소프트웨어공학과  
조교수
- 2016: KAIST 공학박사
- 2012: KAIST 및 CMU 공학석사
- Email: duksan.ryu@jbnu.ac.kr

※ 이 논문은 과학기술정보통신부 및 정보통신기획평가원의 대학CT연구센터지원사업 (IITP-2020-2017-0-01628)과 과학기술정보통신부의 재원으로 한국연구재단(No. NRF-2019R1G1A1005047)의 지원으로 수행된 연구임.