



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

공학석사 학위논문

워크플로우를 지원하는 블록체인  
기반 협업 서비스

BSWork: A Blockchain-based  
Collaborative Service Supporting  
Workflow

울 산 대 학 교 대 학 원

전기전자컴퓨터공학과  
정보통신공학전공

엄 현 민

# 워크플로우를 지원하는 블록체인 기반 협업 서비스

지도교수 이명준

2020년 6월

울산대학교 대학원

전기전자컴퓨터공학과  
정보통신공학전공

엄현민

엄 현 민의 공학석사 학위논문을 인준함

심사위원

권 영 근

(인) 

심사위원

윤 석 훈

(인) 

심사위원

이 명 준

(인) 

울산대학교 대학원

2020년 6월

## [감사의 글]

먼저 석사 과정 동안 부족한 저에게 따뜻한 격려와 조언을 아끼지 않으셨던 이명준 교수님, 박양수 교수님께 감사드립니다. 학부생 시절을 포함하여 4년 동안 이명준 교수님께 받았던 가르침이 있었기에 무사히 석사 과정을 마칠 수 있었습니다. 교수님께서 제가 연구실 생활을 하는 동안 연구에 대한 지도뿐만 아니라 인생에 대한 가르침을 주셨고, 그 덕분에 많이 성장할 수 있었습니다. 또한, 바쁘신 중에도 논문 심사를 맡아주시고 조언을 해주신 권영근 교수님, 윤석훈 교수님께도 정말 감사드립니다.

함께 연구실 생활을 했던 모든 선후배님들께도 감사의 말씀을 드립니다. 연구실 생활에 빠르게 적응할 수 있도록 도와주고 어려움이 있을 때마다 항상 조언을 해주었던 진재환 형, 이우건 형, 함께 밤을 새며 서로의 연구를 도와주고 많은 이야기를 나누었던 윤여국, 조연정, 심채린, 권민호 그리고 항상 격려의 말씀을 해주시던 선배님들 모두 감사드립니다. 프로그래밍 시스템 연구실 식구들과 함께 했던 모든 순간들을 평생 잊지 못할 것 같습니다.

또한, 석사 과정 동안 항상 응원해주신 모든 분들께 감사드립니다. 힘들 때마다 짜증도 많이 냈던 저를 항상 이해해주고 열심히 하는 모습이 멋지다고 말해주던 친구들(이재준, 최은수, 이선우, 김경윤, 어재경), 성공을 위해 빠르게 달려가야 할 바쁜 시기에 졸업 논문을 마무리하느라 적극적으로 참여하지 못한 저를 이해해준 팀원들(이재준, 서강민, 진한성), 그리고 함께 대학원 생활을 하며 즐거운 추억들을 만들어준 장원찬, 최민기, 이경태, 정창홍 형 모두 감사드립니다.

그리고 항상 남들과는 조금 다른 길을 선택하는 저를 믿어주고 응원해주었던 가족들에게 감사의 말씀을 전하고 싶습니다. 떨어져 지내며 연락도 자주 하지 않는 못난 아들을 항상 걱정하며 아낌없는 지원을 해주시던 부모님, 티격태격 자주 싸우지만 누구보다 믿고 의지할 수 있는 든든한 형, 평소에 표현은 못했지만 항상 감사하고 사랑합니다. 또다시 남들과는 조금 다르고 어려운 길을 선택해 걱정이 많으시겠지만, 꼭 성공해서 지금까지 받았던 은혜와 사랑에 몇 배로 보답하겠습니다. 가족들에게 다시 한 번 감사하고 사랑한다는 말을 전합니다.

마지막으로 석사 과정을 무사히 마칠 수 있도록 도와주시고 응원해주신 모든 분들께 다시 한 번 감사드립니다.

## 워크플로우를 지원하는 블록체인 기반 협업 서비스

협업 서비스는 협업 참여자들 간의 원활한 협업 활동을 지원하기 위하여 협업 그룹 생성 및 관리, 멤버들 간의 자료 공유, 그리고 프로젝트 관리 등의 기능을 제공하며, 현재 다양한 분야의 작업환경에서 이러한 협업 서비스를 이용하고 있다. 다수의 참여자가 서로 연관하여 활동하는 협업 환경에서 목적하는 결과물을 차질 없이 도출하기 위해서는 작업 절차가 중요시되고, 참여자들이 이러한 절차와 그에 따른 결과물에 대하여 신뢰할 수 있어야 한다. 따라서 개별 작업의 순서나 작업을 수행하는 대상 등 지정된 작업 절차에 따라 협업 참여자들이 결과물을 도출할 수 있도록 지원하며 이에 대한 신뢰성을 보장하기 위한 방법이 제공되는 것이 바람직하다.

본 논문에서는 미리 정의된 작업 절차를 통하여 신뢰성 있게 작업을 수행하도록 지원하는 블록체인 기반 협업 서비스인 BSWork의 개발에 대하여 기술한다. BSWork는 지정된 작업의 실행 흐름을 위한 워크플로우에 대한 명세를 블록체인 네트워크에 저장하고 이를 통하여 작업의 과정을 제어한다. 이를 위하여, 블록체인의 협업 그룹에 간단하게 작업 과정을 명세할 수 있는 기법과 BSWork에서 블록체인 네트워크 외부와의 상호작용을 지원하기 위한 기법을 제시한다. 이와 더불어, 작업 절차를 통해 도출된 결과물이나 협업 자원과 같은 대용량 데이터를 저장 및 관리하고 이에 대한 신뢰성을 보장하기 위한 오프 블록체인 스토리지 연동 기법을 제시한다.

**주요어** : 워크플로우, 협업 서비스, 블록체인, 이더리움, 분산 어플리케이션

# 목 차

요약 .....	i
목차 .....	ii
그림 목차 .....	iv
표 목차 .....	v
<b>1. 서 론 .....</b>	<b>1</b>
1.1. 연구의 배경 및 필요성 .....	1
1.2. 연구의 개요 .....	3
1.3. 논문의 구성 .....	3
<b>2. 관련 연구 .....</b>	<b>5</b>
2.1 이더리움 블록체인 .....	5
2.1.1 이더리움 블록체인의 개념 .....	5
2.1.2 이더리움 분산 어플리케이션 .....	6
2.2 블록체인 기반 협업 서비스 .....	8
2.3 워크플로우 .....	9
<b>3. BSWork의 정의 .....</b>	<b>10</b>
3.1 BSWork의 구조 .....	10
3.2 BSWork에서의 작업 제어 기법 .....	11
3.2.1 JSFlow .....	11
3.2.2 BSWork에서의 JSFlow 지원 .....	13
3.3 Event-based Function Gateway .....	14

3.4 오프 블록체인 스토리지 연동 .....	15
3.4.1 오프 블록체인 스토리지 연동 구조 .....	15
3.4.2 오프 블록체인 스토리지 연동 기법 .....	17
<b>4. BSWork의 구현 .....</b>	<b>21</b>
4.1 BSWork의 컨트랙트 .....	21
4.2 JSFlow 라이브러리 .....	24
4.3 Gateway 서버 .....	26
4.3.1 BSWork의 Gateway 컨트랙트 .....	26
4.3.2 Gateway 서버 구현 .....	27
4.4 오프 블록체인 스토리지 연동을 위한 REST API 서버 .....	29
4.5 BSWork에서의 참여자 기여도 평가 서비스 .....	30
4.6 분산 어플리케이션 동작화면 .....	33
4.7 BSWork의 유용성 검증 .....	35
<b>5. 결론 .....</b>	<b>37</b>
[참고문헌] .....	38
[Abstract] .....	41



## 그림 목 차

[그림 1] 블록체인의 구조 .....	5
[그림 2] 분산 어플리케이션의 동작 구조 .....	7
[그림 3] BSpace의 이더리움 스마트 컨트랙트 구조 .....	8
[그림 4] BSWork의 구조 .....	10
[그림 5] JSFlow의 구조 .....	12
[그림 6] BSWork에서의 JSFlow 활용 구조 .....	13
[그림 7] Function Gateway 활용 구조 .....	14
[그림 8] 오프 블록체인 스토리지 연동 구조 .....	16
[그림 9] 오프 블록체인 스토리지 기반 파일 업로드 과정 .....	17
[그림 10] 오프 블록체인 스토리지 기반 파일 삭제 과정 .....	20
[그림 11] BSWork의 컨트랙트 구조 .....	21
[그림 12] BSWork에서의 JSFlow 라이브러리 활용 구조 .....	24
[그림 13] JSFlow 라이브러리 사용 예시 .....	26
[그림 14] REST API 서버에서의 파일 접근 권한 검증 구조 .....	30
[그림 15] BSWork에서의 기여도 평가 동작 구조 .....	31
[그림 16] 사용자 등록 화면 .....	33
[그림 17] 그룹 생성 화면 .....	34
[그림 18] 파일 업로드 화면 .....	34
[그림 19] 기여도 평가 화면 .....	35
[그림 20] 설문조사 결과 .....	36

## 표 목 차

[표 1] 사용자 및 그룹 관련 컨트랙트 및 주요 함수 .....	22
[표 2] 협업 그룹 관련 컨트랙트 및 주요 함수 .....	23
[표 3] JSFlow 라이브러리의 주요 함수 .....	25
[표 4] APIConnection 컨트랙트 구조 .....	26
[표 5] APIConnection 컨트랙트의 이벤트 .....	27
[표 6] Gateway 서버 주요 코드 .....	28
[표 7] REST API 서버의 웹서비스 .....	29
[표 8] 기여도 평가를 위한 JSFlow 명세 예시 .....	33

# 1. 서론

## 1.1 연구의 배경 및 필요성

현재 네트워크 환경과 IT기술이 고도화되면서 네트워크를 통한 콘텐츠가 다양해지고 사용자의 콘텐츠의 소비 속도와 양이 빠르게 증가하고 있다. 그에 따라 콘텐츠의 제작자들은 신속하게 양질의 콘텐츠를 생산하기 위해서 제작자들 간의 협업을 수행하고 있으며, 원활한 협업을 위하여 다양한 협업 서비스를 이용하고 있다[1,2,3,4]. 협업 서비스는 다수의 협업 참여자들 간의 효율적인 협업을 지원하기 위한 서비스로서 협업 그룹과 프로젝트의 관리를 위한 기능을 제공한다. 협업 서비스의 사용자들은 제공되는 기능을 통해 협업 그룹을 구성하고, 협업 프로젝트의 자원 및 결과물 등의 협업 데이터를 저장, 공유할 수 있다. 또한, 협업 그룹 구성원에 대한 권한을 지정하고, 지정된 권한 정보를 이용하여 협업 데이터의 저장이나 조회 등의 작업들을 효율적으로 제어할 수 있다.

협업과 같이 다수의 참여자가 서로 연관하여 활동하는 작업 환경에서는 효율적인 작업 수행을 위해 작업 절차가 중요시된다. 협업 환경에서의 작업 절차는 협업을 수행하는 분야의 특성과 협업 그룹 구성원들의 성격에 따라 그 형태가 매우 다양하다. 이러한 절차가 협업 그룹이 목표로 하는 결과물을 차질 없이 도출하기 위해서는 적절한 협업의 절차가 미리 명세되어야 하고, 협업 참여자들이 이러한 절차를 이해하고 동의하면서 협업을 수행할 수 있어야 한다. 이를 위하여, 다수의 협업 그룹들이 지정된 작업의 실행 흐름을 위한 워크플로우에[5] 대한 명세를 이용하여 작업 과정을 제어하고 있으며, 많은 협업 서비스들이 이러한 작업 제어를 지원하는 기능들을 제공하고 있다.

최근에는 중앙 서버를 통해 데이터를 관리하는 서비스들의 보안성에 대한 문제점이 제기되면서 이러한 문제점을 보완하기 위한 기술인 블록체인에[6] 대한 관심이 증가하고 있다. 블록체인 기술은 공개키와 개인키로 구성된 비대칭 키를 이용한 암호화 및 해시함수를 통해 데이터의 위·변조 문제를 보완하고, 이러한 데이터를 분산된 블록체인 노드에 복제 저장하여 데이터의 유실을 방지할 수 있다. 또한, 블록체인 기술이 발전하면서 이더리움 블록체인과 같은 스마트 컨트랙트를[7] 통한 분산어플리케이션(DApp)의 개발이 가능한 프로그래머블 블록체인(Programmable Blockchain)들이 등장하였고[8,9], 그에 따라 다양한 분야에서 블록체인 기반의 어플리케이션에 대한 연구가 활발히 진행되고 있다.

하지만 기존의 협업 서비스들은 중앙 서버를 기반으로 기능을 제공하여 시스템의 실패나 해킹에 의해서 데이터가 유실되거나 위·변조될 수 있다는 치명적인 문제가 발생할 수 있다. 또한, 참여자들 간의 협업 수행의 기록에 대한 신뢰성을 보장할 수 없어 최종 결과물에 대한 기여도 판단에 있어 어려움이 있다. 이러한 문제들로 인하여 참여자들은 자신이 수행한 작업에 대한 기여도를 인정받을 수 없어 소극적으로 협업을 수행하게 되고, 이는 협업 결과물의 품질 저하를 야기할 수 있다. 따라서 협업의 절차, 협업 수행의 기록이나 최종 결과물의 유실 및 위·변조를 방지하여 신뢰성을 보장할 수 있는 방법을 제공하는 것이 바람직하다.

## 1.2 연구의 개요

본 논문에서는 기존의 협업 서비스의 문제점을 해결하기 위하여 미리 정의된 작업 절차를 통하여 신뢰성 있게 작업을 수행하도록 지원하는 블록체인 기반 협업 서비스인 BSWork의 개발에 대하여 기술한다. BSWork는 지정된 작업의 실행 흐름을 위한 워크플로우에 대한 명세와 참여자의 작업 수행에 대한 기록을 블록체인 네트워크에 저장하고, 저장된 워크플로우에 대한 명세를 통한 작업 제어 기능을 제공하여 효율적인 협업 수행을 지원한다. 본 논문에서는 BSWork를 이더리움 블록체인 기반의 분산 어플리케이션으로 구현하여 기능의 유효성을 검증한다.

구현된 BSWork는 이더리움 스마트 컨트랙트를 이용하여 협업 결과물 및 자원 등의 협업 데이터 관리 기능과 워크플로우에 대한 명세 저장 및 이를 통한 작업 제어 기능을 제공한다. 이를 위하여, 본 논문에서는 JSON을 이용하여 간단하게 작업 과정을 명세할 수 있는 기법과 이를 스마트 컨트랙트에서 활용하기 위한 이더리움 라이브러리에 대하여 기술한다. 또한, 다양한 협업 환경에서 스마트 컨트랙트의 함수만으로는 원활한 협업 진행을 지원하는데 어려움이 있어 블록체인 네트워크 외부의 추가적인 모듈 사용이 불가피하므로 이를 지원하기 위한 방안인 Function Gateway 컨트랙트를 제안한다. 제안하는 Function Gateway 컨트랙트는 이더리움 스마트 컨트랙트와 이더리움 분산 어플리케이션에서 활용하는 Event-Watch 기능을 이용하여 외부 모듈의 함수를 호출한다. 또한, Function Gateway 컨트랙트는 외부 모듈의 함수 호출 과정이 블록체인에 기록되어 BSWork에서 해당 모듈을 사용하는 작업의 과정에 대한 신뢰성을 보장할 수 있다는 장점을 가진다.

이와 더불어, 구현된 BSWork는 저장 가능한 데이터의 크기가 제한된 블록체인에서 워크플로우에 나타나는 협업 결과물 및 자원과 실제 협업 데이터간의 관계를 성립하기 위하여 오프 블록체인 스토리지에 협업 데이터를 저장한다. 이를 위하여, 본 논문에서는 Function Gateway 컨트랙트를 통한 오프 블록체인 스토리지 연동 기법에 대하여 설명한다. 또한, BSWork의 응용 서비스로 협업 참여자의 결과물에 대한 기여도를 평가하기 위한 기여도 평가 서비스를 개발하고 이를 통해 기능의 유효성을 확인하였다.

### 1.3 논문의 구성

논문의 구성은 다음과 같다. 다음 장에서는 배경지식으로 이더리움 분산 어플리케이션과 블록체인 기반 협업 서비스 그리고 워크플로우에 대해서 다루고 3장에서는 BSWork에 대하여 정의한다. 또한, 4장에서는 BSWork의 구현에 대하여 설명하고 BSWork의 응용 서비스인 기여도 평가 서비스에 대하여 기술한다. 마지막으로 5장에서는 결론을 제시한다.

## 2. 관련 연구

본 장에서는 이더리움 블록체인 및 이더리움 분산 어플리케이션의 개념과 블록체인 기반 협업 서비스에 대하여 기술한다. 또한, 워크플로우의 특징과 대표적인 워크플로우 명세를 위한 방법에 대하여 설명한다.

### 2.1 이더리움 블록체인

이더리움은 2세대 블록체인 기술로서 다수의 블록체인 노드로 구성된 블록체인 네트워크를 기반으로 분산 어플리케이션(Decentralized Application, DApp)을[10] 개발 및 구동하기 위한 플랫폼이다. 이더리움 분산 어플리케이션은 이더리움 블록체인 네트워크에 배포된 스마트 컨트랙트(Smart Contract)를 이용하여 데이터를 저장하고 블록체인에서 제공하는 기능을 실행하기 위한 트랜잭션을 발생시킨다.

#### 2.1.1 이더리움 블록체인의 개념

블록체인은 최초의 암호 화폐이자 1세대 블록체인인 비트코인[11]의 기반 기술로서 데이터의 위·변조를 방지하기 위해 p2p기반 분산 환경에 데이터를 저장하는 기술이다. 그림 1은 블록체인 기술의 기본적인 구조에 대하여 나타낸다.

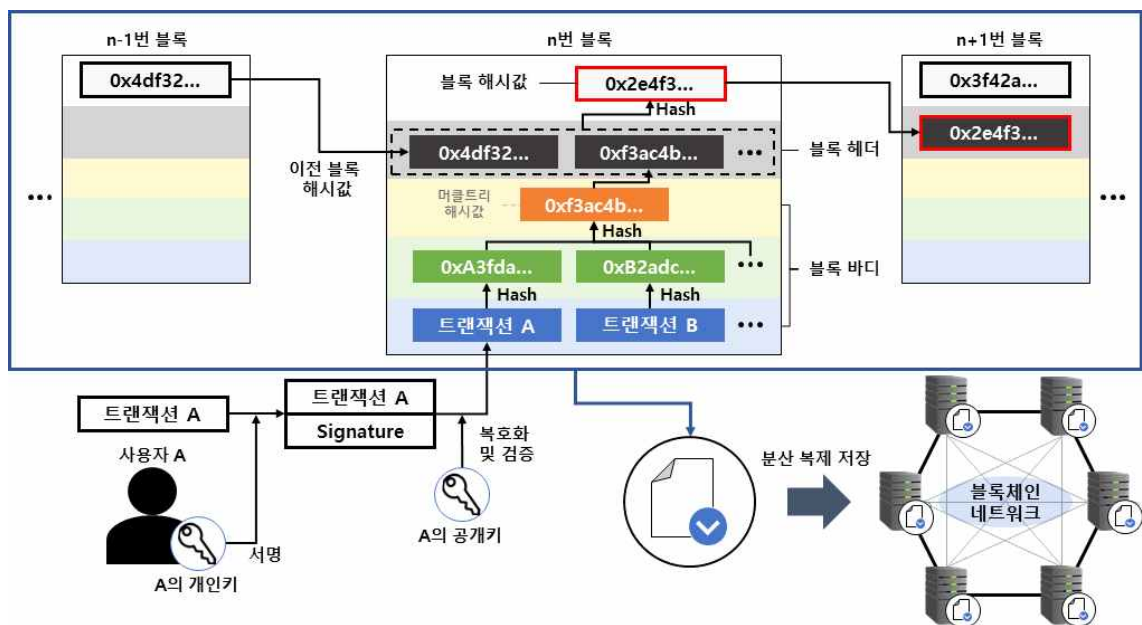


그림 2 블록체인의 구조

블록체인 기술의 블록은 저장되는 데이터들의 집합을 의미하며, 이러한 블록 단위의 데이터 집합들은 블록체인 네트워크를 구성하는 분산 노드들에 서로 연관된 체인 형태로 복제되어 저장된다. 데이터의 저장 등의 블록체인 네트워크의 기능들은 분산 노드를 통해 전달된 트랜잭션 기반의 요청에 의해 수행되며, 이러한 트랜잭션에 대한 기록 또한 블록에 저장된다. 블록체인 기술은 전달된 트랜잭션의 검증 및 저장되는 데이터의 무결성을 보장하기 위해 개인키/공개키를 사용하는 비대칭 키 암호화 방식과 임의의 길이의 데이터를 고정된 길이의 데이터로 매핑하는 해시함수를 사용한다.

이더리움 블록체인은 이러한 블록체인 기술이 발전된 2세대 블록체인으로서 스마트 컨트랙트를 통한 분산 어플리케이션의 개발이 가능한 프로그래머블 블록체인(Programmable Blockchain)이다. 이더리움의 스마트 컨트랙트는 비트코인의 스크립트와 달리 상태 변수를 저장할 수 있고 이를 조회하거나 활용하는 함수를 작성할 수 있다. 또한, 함수 실행가드인 Modifier를 통하여 특정 조건에 의해 함수가 실행되도록 제어할 수 있고, 이벤트를 통하여 블록체인 네트워크 외부로 데이터를 전달할 수 있다. 이러한 스마트 컨트랙트는 전용 프로그래밍 언어인 Solidity를[12] 통해 작성되며, 작성된 스마트 컨트랙트는 블록체인 네트워크에 배포되어 분산 어플리케이션에서 필요로 하는 기능의 수행을 지원한다.

이더리움 블록체인의 사용자는 스마트 컨트랙트의 함수 호출이나 이더리움 블록체인인 암호화체인 이더를 전송하기 위한 트랜잭션을 블록체인 네트워크에 전달하기 위해 자신의 어카운트를 사용한다. 이더리움 어카운트는 사용자를 식별하기 위한 식별 값인 어카운트 어드레스를 가지고 있으며, 하나의 어카운트에는 개인키와 공개키로 구성된 비대칭 키가 포함되어 있다. 이러한 비대칭 키는 블록체인 네트워크에 전달된 트랜잭션이 해당 사용자에 의해 생성되었음을 증명하기 위한 서명(Sign)에 사용된다. 트랜잭션에 대한 서명은 수행하고자 하는 기능의 정보가 포함된 트랜잭션 메타데이터를 해당 기능을 수행하는 사용자의 개인키로 암호화하는 것을 의미하며 이러한 서명 과정은 암호화 된 트랜잭션 메타데이터인 서명 정보(Signature)를 반환한다. 서명 정보는 트랜잭션 메타데이터와 함께 블록체인 네트워크로 전달되며, 전달된 서명 정보는 서명한 사용자의 공개키로만 복호화가 가능하고 서명 정보와 암호화 전의 데이터를 통해 서명자의 어카운트 어드레스를 확인할 수 있어 해당 트랜잭션에 대한 검증에 사용된다.

### 2.1.2 이더리움 분산 어플리케이션

이더리움 분산 어플리케이션은 이더리움 블록체인 네트워크에 배포된 스마트 컨트



랙트를 기반으로 동작하는 어플리케이션이다. 분산 어플리케이션은 블록체인 노드에서 제공하는 API를 통하여 스마트 컨트랙트에 정의된 함수 및 변수를 활용한 다양한 기능을 제공할 수 있고, 이를 위하여 블록체인 노드의 API 사용을 지원하는 Web3 라이브러리를[13] 이용한다. 그림 2는 이더리움 분산 어플리케이션의 동작 구조를 나타낸다.

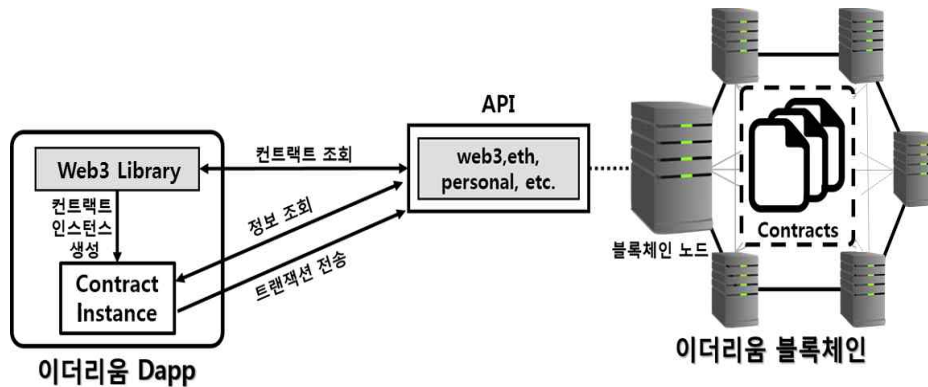


그림 3 분산 어플리케이션의 동작 구조

분산 어플리케이션은 Web3 라이브러리를 통해 블록체인 네트워크에 배포된 스마트 컨트랙트와 연관된 객체를 생성하고 이를 활용하여 컨트랙트의 함수를 호출한다. 분산 어플리케이션에서 스마트 컨트랙트와 상호작용하기 위한 과정은 다음과 같다.

- (1) Web3 라이브러리를 통하여 사용하고자 하는 스마트 컨트랙트의 정보를 조회한다.
- (2) 조회된 컨트랙트의 정보와 컨트랙트의 기본적인 구조에 대하여 명세된 ABI를 통하여 실제 컨트랙트와 연관된 컨트랙트 객체를 생성한다.
- (3) 생성된 컨트랙트 객체를 통하여 호출하고자 하는 함수와 관련된 파라미터를 바탕으로 트랜잭션을 생성한다.
- (4) Web3 라이브러리를 통해 생성된 트랜잭션을 블록체인 노드로 전달하여 함수를 실행한다.

분산 어플리케이션의 기능들은 트랜잭션을 통해 수행되고 이더리움 블록체인 네트워크에서는 전달받은 모든 트랜잭션을 다수의 노드에서 복제하고 있으므로 분산 어플리케이션의 기능 수행 내역을 안정적으로 관리할 수 있게 된다. 이와 같은 특징으로, 현재 이더리움 분산 어플리케이션은 신뢰성 있는 서비스 제공을 필요로 하는 다양한 분야에서 많이 활용되고 있다[14].

## 2.2 블록체인 기반 협업 서비스

최근 기존 협업 서비스의 데이터 유실이나 위·변조 등과 같은 신뢰성의 문제가 이슈화되면서 이를 해결하기 위한 방안으로 블록체인 기반의 협업 서비스에 대한 연구가 활발히 진행되고 있다. 블록체인 기반 협업 서비스는 블록체인 네트워크에 배포되어 있는 스마트 계약을 통해 협업 그룹이 필요로 하는 기능들은 지원하며 이를 통해, 기존 협업 서비스의 신뢰성 문제를 보완한다. 이를 위하여, 블록체인 기반 협업 서비스는 스마트 계약을 통해 협업 그룹과 협업 그룹의 프로젝트를 관리하기 위한 기능을 지원한다. 대표적인 블록체인 기반 협업 서비스로는 최근 연구결과가 발표된 BSpace가 있다[15].

### 1) BSpace

BSpace는 이더리움 블록체인 기반의 협업 서비스이며 협업 그룹과 그룹 멤버, 협업 자원 관리 등의 협업 기능을 제공하기 위하여 이더리움 스마트 계약을 이용한다. 이러한 BSpace의 주요 스마트 계약에는 협업 그룹의 생성과 그룹 멤버의 등록 및 권한을 관리하기 위한 UserGroup 계약과 협업 그룹의 프로젝트 및 이와 관련된 자원 및 결과물의 저장, 공유 등의 관리를 위한 Workspace 계약 등이 있다. 그림 3은 BSpace의 이더리움 스마트 계약 구조를 나타낸다.

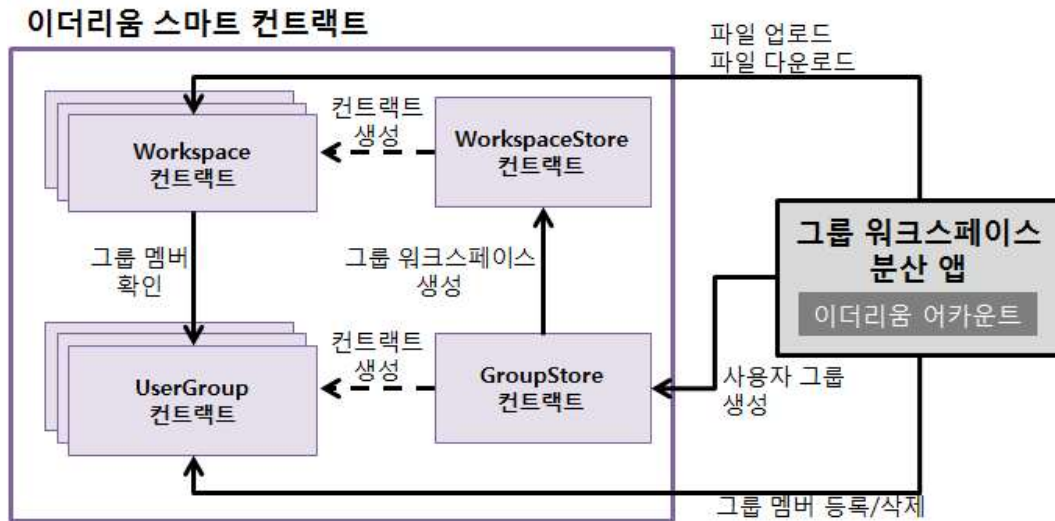


그림 4 BSpace의 이더리움 스마트 계약 구조

BSpace를 위한 분산 어플리케이션은 웹 기반으로 구현되었으며, 사용자 인터페이

스를 통해 스마트 컨트랙트의 기능을 간편하게 실행할 수 있도록 지원한다. 또한, 구현된 이더리움 분산 어플리케이션의 사용자는 자신의 이더리움 어카운트 BSpace의 기능을 실행할 수 있다.

## 2.3 워크플로우

워크플로우는 작업에 대한 일련의 흐름으로서 작업의 순서 및 작업 수행 대상 등의 작업 과정에 대한 다양한 정보를 포함하고 있다. 다수의 참여자로 구성된 작업환경에서는 작업의 절차가 효율적인 작업을 위한 중요한 요소이며, 참여자들의 작업이 이러한 절차를 따라 수행될 수 있도록 제어되어야 한다. 또한, 작업 환경의 작업 절차가 복잡하고 다양해질수록 이를 효과적으로 제어할 수 있어야 한다. 이를 위하여 많은 분야의 작업환경에서 작업 과정을 제어할 위하여 지정된 작업의 실행 흐름을 위한 워크플로우에 대한 명세를 이용하고 있다. 워크플로우에 대한 명세를 통한 작업 제어를 작업 환경에 적용함으로써 얻게 되는 이점은 다음과 같다[16].

- 작업의 효율성을 증가시킬 수 있다.
- 작업의 실행 흐름을 일관적이고 체계적으로 관리할 수 있다.
- 작업의 복잡성 증가에 대하여 효과적인 대응이 가능하여 작업의 확장성이 용이해질 수 있다.
- 작업의 신속한 수행이 가능해지고, 그에 따라 도출된 결과물의 품질을 향상시킬 수 있다.

워크플로우에 대한 명세를 위한 방법에는 다양한 형태들이 있으며 대표적으로 DCR Graph, State Chart 등이 있다[17,18]. 이러한 워크플로우에 대한 명세를 위한 방법들은 작업환경에 따라 사용하는 프레임워크와 프로그래밍 언어 등이 상이하기 때문에 다양한 환경에서 이를 지원하기 위한 도구들이 개발되고 있다. 최근 블록체인 기반의 서비스들이 빠르게 증가됨에 따라 Yakindu에서는 이더리움 스마트 컨트랙트 간편한 작성을 지원하기 위하여 State Chart를 기반으로 이더리움 스마트 컨트랙트의 기본적인 구조를 자동으로 생성하는 서비스에 대한 연구가 진행 중이다[19].

### 3. BSWork의 정의

본 장에서는 워크플로우를 지원하는 블록체인 기반 협업 서비스인 BSWork에 대하여 설명한다. 또한, BSWork를 효율적으로 제공하기 위한 여러 기법들에 대하여 기술한다.

#### 3.1 BSWork의 구조

기존의 협업 서비스들은 협업과 관련된 결과물 및 자원이나 협업 수행의 기록에 대한 신뢰성을 보장할 수 없어 협업의 참여자들이 최종 결과물에 대한 적절한 기여도를 인정받지 못하는 문제가 발생할 수 있다. 이로 인하여 협업의 참여자들은 기존의 협업 서비스를 이용하는 작업 환경에서 적극적인 협업을 수행하기 어렵다. 따라서 본 절에서는 협업 자원이나 최종 결과물의 유실 및 위·변조를 방지하고, 협업의 절차 및 협업 수행의 기록의 신뢰성을 보장할 수 있도록 지원하는 블록체인 기반 협업 서비스인 BSWork에 대하여 설명한다.

BSWork는 이더리움 블록체인을 기반으로 협업 참여자들의 원활한 협업 활동을 지원하는 서비스이다. BSWork의 기본적인 구조에는 협업 그룹 및 협업 데이터에 대한 정보를 저장하기 위한 이더리움 스마트 컨트랙트와 이를 활용하는 분산 어플리케이션이 포함된다. 또한, 실질적인 협업 데이터를 저장하기 위한 블록체인 외부의 스토리지인 오프 블록체인 스토리지와 이를 이더리움 스마트 컨트랙트와 연동하기 위한 Gateway 서버 및 REST API 서버로 구성된다. 그림 4는 BSWork의 기본적인 구조에 대하여 나타낸다.

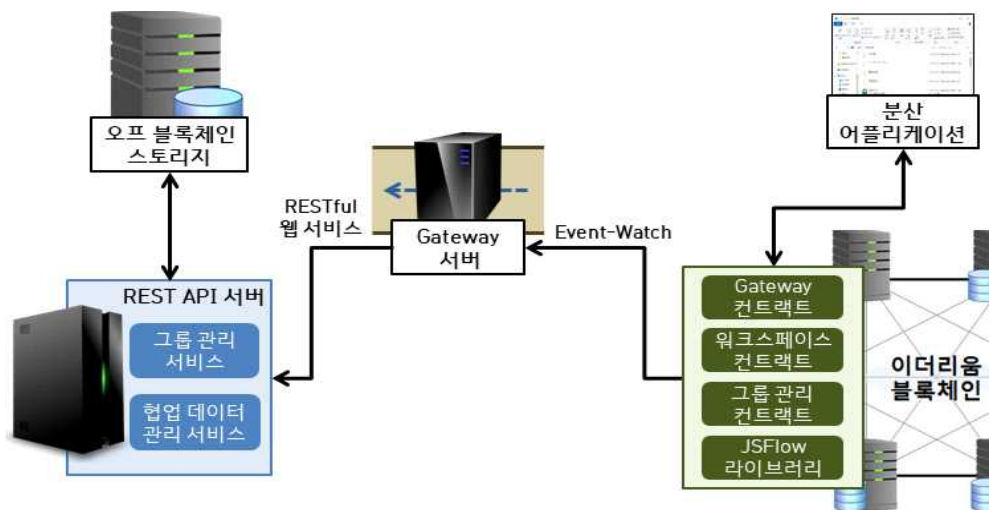


그림 5 BSWork의 구조

BSWork의 사용자들은 제공되는 분산 어플리케이션을 통해 그룹을 생성하거나 협업 데이터를 저장 및 공유할 수 있고 이에 대한 정보들은 각각 그룹 관리 컨트랙트와 워크스페이스 컨트랙트에 저장된다. BSWork에서는 이후 설명하는 JSFlow와 JSFlow 라이브러리를 통하여 협업 그룹의 워크플로우에 대한 명세와 이를 통한 작업 제어 기능을 제공한다. 또한, BSWork는 워크플로우에 나타나는 협업 결과물 및 자원과 실제 협업 데이터간의 관계를 성립하기 위하여 오프 블록체인 스토리지에 실제 협업 데이터를 저장한다. REST API 서버는 오프 블록체인 스토리지에 저장된 협업 데이터에 대한 업로드, 다운로드 등의 서비스를 제공한다. 이와 더불어, Gateway 서버는 REST API 서버의 서비스와 같이 블록체인 외부의 기능의 사용을 지원하고, Gateway 컨트랙트를 통하여 이에 대한 신뢰성을 보장한다.

## 3.2 BSWork에서의 작업 제어 기법

본 절에서는 BSWork에서 워크플로우에 대한 명세를 통한 작업 제어를 제공하기 위한 방법에 대하여 설명한다. 이를 위하여 간단하게 작업 과정을 명세할 수 있는 기법과 이를 스마트 컨트랙트에서 활용하기 위한 기본적인 구조에 대하여 기술한다.

### 3.2.1 JSFlow

다수의 블록체인 기반 서비스에서는 작업 절차에 따라 작업을 제어하기 위하여, 워크플로우에 대한 명세를 기반으로 스마트 컨트랙트를 작성하거나 블록체인 외부에서 지정된 절차에 따라 스마트 컨트랙트의 함수 호출하여 작업을 제어하는 등의 방식을 사용하고 있다. 하지만 이러한 방식은 워크플로우에 대한 명세 과정이 복잡하고 이와 같은 워크플로우를 블록체인 환경에서 지원하기 위해서는 많은 비용이 발생한다. 소규모의 협업 그룹에서는 복잡한 워크플로우를 사용하는 경우가 드물기 때문에 이러한 복잡한 워크플로우 명세 방법을 사용하기에는 어려움이 있다. 스마트 컨트랙트의 특성상 한번 작성되어 블록체인 네트워크에 배포되면 수정이 불가능하여 적용 분야와 작업 과정이 변경됨에 따라 새로운 컨트랙트에 대한 구현 및 배포로 인한 추가적인 비용 발생하게 된다. 또한, 블록체인 외부에서 작업을 제어하는 방식의 경우 작업 과정에 대한 신뢰성 보장이 어려운 문제점이 있다.

본 논문에서는 이러한 문제점들에 대한 해결 방안으로 간단하게 유동적인 작업 과정에 대한 명세를 할 수 있는 기법인 JSFlow에 대하여 제안한다. 제안하는 JSFlow는 작업의 절차에 대한 워크플로우 명세를 위하여 JSON 형식을 이용한다. 이를 위한 기

본적인 형태의 JSON에는 작업과 해당 작업의 과정에 대한 정보가 각각 key-value 쌍으로 저장된다. 그림 5는 제안하는 JSFlow의 기본적인 구조에 대하여 나타낸다.

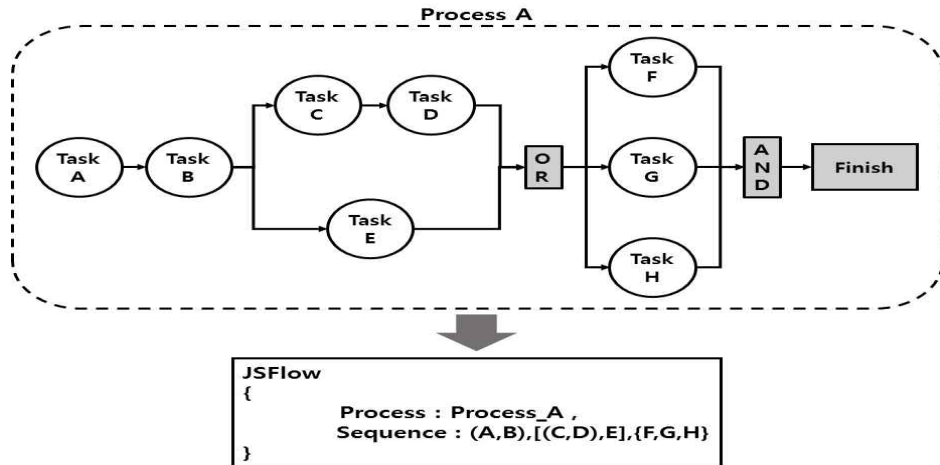


그림 6 JSFlow의 구조

그림 1의 작업 과정의 동작 수행순서는 다음과 같다.

- (1) 동작 A를 수행한 후 동작 B를 수행
- (2) “동작 C를 수행한 후 동작 D를 수행” 또는 “동작 E를 수행”
- (3) 순서에 관계없이 동작 F, G, H 모두 수행

작업 과정 정보에 저장된 수행동작들의 기본적인 수행순서는 순차적 수행이며, 작업 과정의 특성에 따라 “( )”, “[ ]”, “{ }” 와 같은 특정 기호를 이용하여 동작들의 수행순서를 지정할 수 있다. 수행순서를 지정하기 위한 기호들은 각각 다음을 의미한다.

- 1) “( )” - 지정된 동작들을 순차적으로 수행(Sequence)
- 2) “[ ]” - 지정된 동작들 중 하나의 동작 수행(Or)
- 3) “{ }” - 지정된 동작들을 순서에 관계없이 모두 수행(And)

이러한 특정 기호들은 다른 기호와 조합하여 사용할 수 있으며, 이를 활용하여 제안된 JSFlow는 간단하게 다양한 형태의 작업 절차에 대한 명세를 지원할 수 있다. JSFlow는 협업 그룹이 사용하는 협업 서비스에서 간단한 명세를 통해 워크플로우를 지원하므로 소규모의 협업 그룹에서는 간단한 워크플로우에 대한 명세를 위하여 JSFlow를 사용하는 것이 효율적이다.

### 3.2.2 BSWork에서의 JSFlow 지원

BSWork는 JSFlow 명세를 활용한 작업 과정 제어를 위하여 협업 데이터의 관리를 지원하는 워크스페이스 컨트랙트에 JSFlow 명세를 협업 데이터의 형태로 저장한다. 그림 6은 BSWork에서의 JSFlow 활용 구조를 나타낸다.

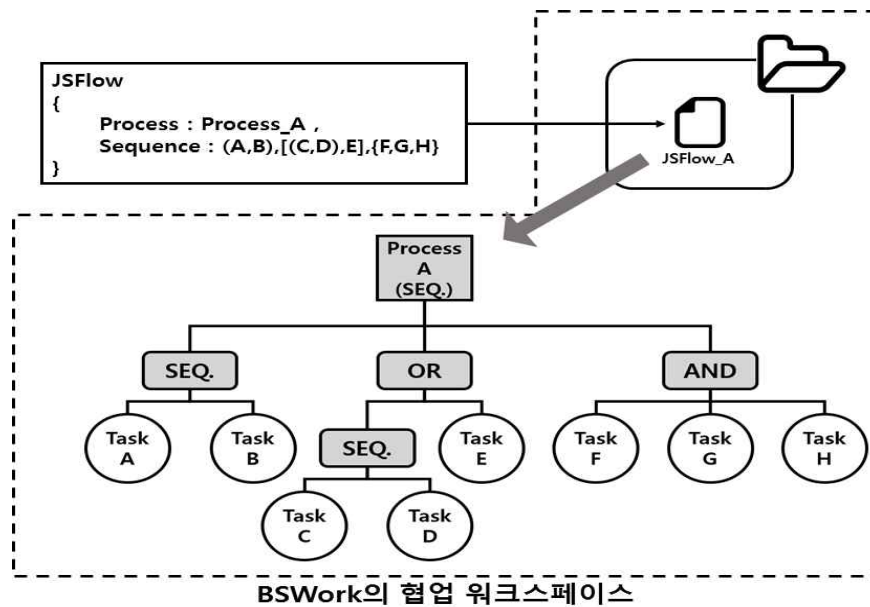


그림 7 BSWork에서의 JSFlow 활용 구조

BSWork는 저장된 JSFlow 명세를 기반으로 AND-OR 트리를 생성하고 이를 통하여 작업과정을 제어한다. AND-OR 트리를 구성하는 노드에는 동작의 수행흐름에 대하여 정의된 노드 및 수행 동작에 대하여 정의된 노드와 같은 다수의 노드들이 포함된다. 동작의 수행 흐름에 대하여 정의된 노드는 SEQ. 노드(순차 수행), AND 노드(동시 수행), OR 노드(선택 수행) 등으로 구분된다. 또한, 이러한 수행 흐름 노드들은 자신의 자식 노드로 다른 수행흐름 노드 및 수행 동작에 대하여 정의된 노드인 LEAF 노드를 가질 수 있다. SEQ. 노드의 경우 동작 수행 흐름은 자식 노드들 중 가장 좌측의 노드부터 우측의 노드까지 순차적으로 진행된다. 트리의 최상위 노드인 root 노드는 SEQ. 노드이며, 수행 흐름 노드들만이 ROOT 노드의 자식노드가 될 수 있다. BSWork는 생성된 AND-OR 트리를 기반으로 현재 수행 가능한 동작에 대한 정보를 조회하고 이를 제어하기 위하여 해당 동작에 대한 LEAF 노드들의 정보를 저장한다.

### 3.3 Event-based Function Gateway

이더리움 분산 어플리케이션은 스마트 컨트랙트의 함수를 통하여 블록체인에 데이터를 저장 및 조회하는 등의 기능을 수행할 수 있다. 이더리움 스마트 컨트랙트의 함수들은 트랜잭션을 통한 요청에 의해 호출되고 이러한 트랜잭션에 대한 기록들은 블록체인에 저장된다. 하지만 활용 분야에 따라 분산 어플리케이션의 기능이 다양해지고 복잡해지면서 스마트 컨트랙트의 함수만으로 제공하고자 하는 기능을 지원하기에는 어려움이 있다. 또한, 이러한 기능들을 지원하기 위하여 외부 모듈을 사용하는 경우 블록체인에 저장된 데이터를 활용하기 어렵고, 기능 수행에 대한 기록을 블록체인에 저장할 수 없어 신뢰성을 보장할 수 없다.

본 절에서는 이더리움 분산 어플리케이션에서 블록체인 외부의 모듈에 대한 사용을 지원하고 이에 대한 신뢰성을 보장하기 위하여 Function Gateway 활용 기법을 제안한다. 제안하는 기법에서는 외부 모듈의 기능을 제공하기 위한 별도의 서버인 Gateway 서버를 이용하고 분산 어플리케이션과 Gateway 서버 간의 원활한 상호작용을 지원하기 위하여 Function Gateway 컨트랙트를 사용한다. Function Gateway 컨트랙트는 이더리움 스마트 컨트랙트의 Event-Watch 처리기법을 이용하여 분산 어플리케이션과 Gateway 서버를 연결하는 역할을 한다. 이더리움 스마트 컨트랙트의 Event-Watch 처리기법은 스마트 컨트랙트에서 외부로 데이터를 전달하기 위해 발생하는 이벤트와 이러한 이벤트를 Web3 라이브러리를 활용하는 어플리케이션에서 수신하여 이벤트에 저장된 데이터를 조회하는 과정이다. 그림7은 Function Gateway를 이용하는 분산 어플리케이션의 구조에 대하여 나타낸다.

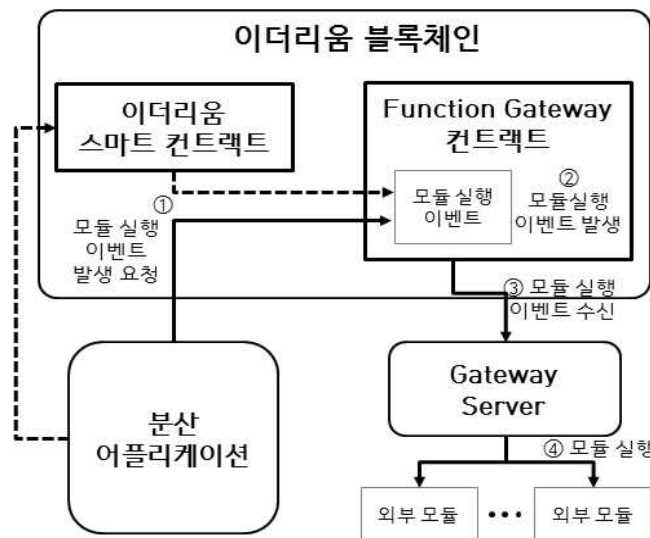


그림 8 Function Gateway 활용 구조



제안된 기법을 위한 Function Gateway 컨트랙트에는 분산 어플리케이션에서 필요로 하는 외부 모듈의 기능의 정보 및 관련된 메타데이터를 전달하기 위한 이벤트와 이러한 이벤트를 발생시키는 함수가 존재한다. 분산 어플리케이션은 트랜잭션을 통한 직접적인 요청 또는 다른 이더리움 컨트랙트를 이용한 요청을 통해 Function Gateway 컨트랙트의 이벤트 발생 함수를 호출할 수 있다. Function Gateway 컨트랙트를 통해 외부 모듈을 사용하는 과정은 다음과 같다.

- (1) 분산 어플리케이션 또는 분산 어플리케이션의 요청을 받은 이더리움 컨트랙트는 모듈의 실행을 위해 Function Gateway 컨트랙트의 모듈 실행 이벤트 발생 함수를 호출한다.
- (2) Function Gateway 컨트랙트는 호출된 함수의 파라미터를 통하여 실행하고자 하는 외부 모듈과 연관된 이벤트를 발생한다.
- (3) Gateway 서버는 수신한 모듈 실행 이벤트에 따라 필요 모듈의 기능 수행을 요청한다.

Function Gateway 컨트랙트를 통한 외부 모듈 기능의 사용은 이를 위한 요청이 트랜잭션을 통하여 이루어지므로 기능의 수행에 대한 기록이 블록체인에 저장되어 신뢰성을 보장할 수 있다. 또한, 다른 이더리움 컨트랙트를 통해 Function Gateway 컨트랙트에 요청하는 경우에는 해당 컨트랙트에 저장된 데이터를 이용하거나 호출되는 외부 모듈과 관련된 데이터를 저장할 수 있어 효율적인 기능 수행이 가능하다.

### 3.4 오프 블록체인 스토리지 연동

본 절에서는 앞에서 설명한 Function Gateway를 활용하여 대용량 데이터를 블록체인 외부의 스토리지에 저장하기 위한 오프 블록체인 스토리지 연동 기법에 대하여 제안한다.

#### 3.4.1 오프 블록체인 스토리지 연동 구조

이더리움 스마트 컨트랙트는 한 번의 요청으로 전송할 수 있는 데이터의 용량에 제한이 있어 협업 결과물이나 자원 등의 실질적인 데이터를 저장하기에는 어려움이 있다. BSWork는 이러한 한계점을 보완하기 위하여 제안하는 오프 블록체인 스토리지 연동 기법을 통해 오프 블록체인 스토리지에 협업 데이터를 저장한다. 제안하는 기법에서는 이더리움 블록체인 네트워크의 스마트 컨트랙트와 오프 블록체인 스토리지를

연동하기 위하여 웹 서비스를 제공하는 REST API 서버와 Gateway 서버를 이용한다. 그림 8은 BSWork에서 제안하는 기법을 기반으로 오프 블록체인 스토리지와 연동하기 위한 구조에 대하여 나타낸다.

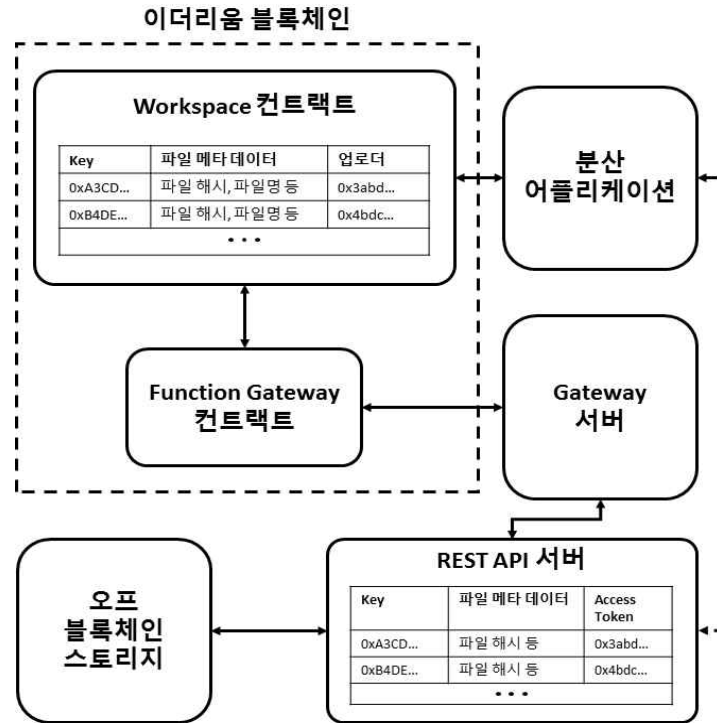


그림 9 오프 블록체인 스토리지 연동 구조

BSWork의 워크스페이스 컨트랙트는 오프 블록체인 스토리지에 저장될 협업 데이터의 위·변조 여부를 확인하기 위하여 분산 어플리케이션에서 전달된 협업 데이터의 해시값, 작성자, 데이터 크기 등의 정보를 저장한다. Function Gateway 컨트랙트는 워크스페이스 컨트랙트를 통해 받은 협업 데이터에 대한 요청을 받고, 이를 블록체인 네트워크 외부로 전달하기 위한 역할을 한다. Gateway 서버는 Event-watch 기능을 이용하여 Function Gateway 컨트랙트에서 발생한 이벤트를 수신하고 이를 REST API 서버로 전달한다. REST API 서버는 오프 블록체인 스토리지에 접근하는 요청에 대하여 실제 데이터를 송수신하는 역할을 수행하기 위한 RESTful 웹서비스 제공 서버이다. 또한, REST API 서버는 워크스페이스 컨트랙트에 기록된 협업 데이터의 정보와 실제 협업 데이터가 일치함을 검증하기 위하여 실제 데이터의 정보를 저장한다. 오프 블록체인 스토리지는 실제 협업 데이터를 저장하며, 보안을 위해 REST API 서버에서만 접근이 가능하다.

### 3.4.2 오프 블록체인 스토리지 연동 기법

본 항에서는 BSWork에서 오프 블록체인 스토리지에 저장된 협업 데이터 파일에 관련된 기능을 수행하기 위한 과정을 설명한다. BSWork의 분산 어플리케이션은 파일 업로드 시 업로드하고자 하는 파일의 메타 정보를 확인하고 해당 파일에 대한 해시 값을 생성한다. 그 후, 분산 어플리케이션은 업로드할 파일에 대한 메타 정보와 해시 값을 워크스페이스 컨트랙트의 파일 업로드 함수를 통해 전달하고 파일 업로드 함수에서는 전달받은 정보를 스마트 컨트랙트에 저장한다. 그림 9는 파일 업로드를 위한 과정을 나타낸다.

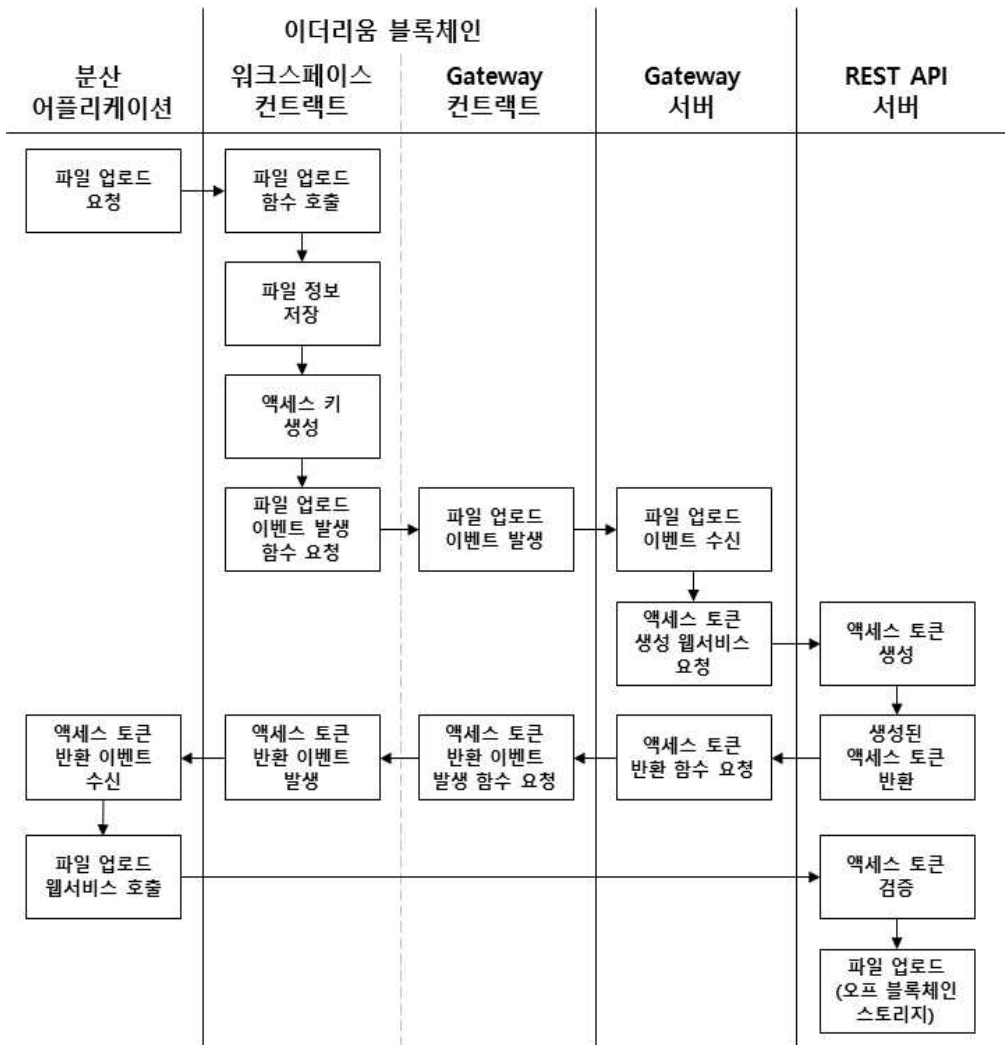


그림 10 오프 블록체인 스토리지 기반 파일 업로드 과정

파일의 정보에는 파일명 및 파일 사이즈, 확장자, 공유시각 및 공유자 정보 등이 포

함된다. 업로드 파일에 대한 정보와 해시 값은 Workspace 컨트랙트 내에 저장되며 저장 후 REST API 서버로 전달하기 위한 과정을 수행한다. 분산 어플리케이션에서 업로드하고자 하는 파일이 오프 블록체인 스토리지에 저장되기 위한 과정은 다음과 같다.

- (1) 분산 어플리케이션의 사용자는 업로드하고자 하는 파일을 분산 어플리케이션에 등록한다.
- (2) 분산 어플리케이션은 등록된 파일의 메타 정보를 확인하고 파일에 대한 해시값과 해시값을 사용자의 개인키로 서명한 서명정보를 생성한다.
- (3) 분산 어플리케이션은 생성된 해시값과 서명정보 및 파일의 메타 정보를 파라미터로 워크스페이스 컨트랙트의 파일 업로드 함수를 호출한다.
- (4) 워크스페이스 컨트랙트는 호출된 파일 업로드 함수를 통해 전달받은 서명정보를 포함한 파일에 대한 정보들을 저장하고, 이를 통해 파일에 접근하기 위한 액세스 키를 생성하여 저장한다.
- (5) 워크스페이스 컨트랙트는 파일의 해시값, 서명정보, 그리고 생성된 액세스 키를 파라미터로 하여 Gateway 컨트랙트의 파일 업로드 이벤트 발생 함수를 호출한다.
- (6) Gateway 컨트랙트는 호출된 파일 공유 이벤트 발생 함수를 통해 전달받은 파라미터를 포함하여 파일 공유 이벤트를 발생시킨다.
- (7) Gateway 서버는 발생한 이벤트를 수신하고 수신한 이벤트에 포함된 정보를 REST API 서버의 액세스 토큰 생성을 위한 웹 서비스 요청을 통해 전달한다.
- (8) REST API 서버는 전달된 파일의 해시값과 서명정보를 통해 사용자의 어카운트 어드레스를 확인한 후 이를 포함한 정보들을 통해 파일 업로드 웹 서비스 요청을 위한 액세스 토큰 생성하고 관련된 정보들을 저장한다.
- (9) REST API 서버는 Gateway 서버를 통해 생성된 액세스 토큰을 파라미터로 하여 Gateway 컨트랙트의 액세스 토큰 반환 함수를 호출한다.
- (10) Gateway 컨트랙트는 워크스페이스 컨트랙트의 액세스 토큰 반환 이벤트 발생 함수를 호출하여 전달받은 액세스 토큰을 포함한 이벤트를 발생시킨다.
- (11) 분산 어플리케이션은 발생한 이벤트를 수신하고 수신한 이벤트에 포함된 액세스 토큰과 공유하고자 하는 파일 그리고 액세스 토큰을 사용자의 개인키로 서명한 서명 정보를 REST API 서버의 파일 공유를 위한 웹 서비스 요청을 통해 전달한다.
- (12) REST API는 전달받은 액세스 토큰 및 서명 정보를 통해 확인된 어카운트 어드레스와 저장된 사용자 어카운트를 비교하고, 전달된 파일에 대한 해시값과 저장된 해시값을 비교하여 사용자와 파일에 대한 검증을 완료한 후 오프 블록체인 스토리지에 해당 파일을 액세스 키와 함께 저장한다.

이러한 과정을 통해 블록체인 네트워크에서 발생된 요청을 외부로 전달할 수 있으며 이를 위한 과정에 대한 기록들은 스마트 컨트랙트를 통해 블록체인 네트워크에 저장된다. 또한, 액세스 토큰과 서명 정보 등을 통해 블록체인 네트워크 외부에서도 사용자의 파일 접근에 대한 권한 인증 처리가 가능해지고, 액세스 토큰을 사용할 수 있는 제한된 유효 시간을 지정하여 액세스 토큰의 유출 등에 문제로 인한 무분별한 파일 접근을 방지할 수 있다. 이와 더불어, 웹 서비스를 통해 받은 파일에 대한 해시값 생성하고 이를 스마트 컨트랙트를 통해 전달받은 파일 해시값과의 일치 여부를 확인하여 파일의 위·변조가 없음을 확인할 수 있다.

또한, BSWork의 분산 어플리케이션은 오프 블록체인 스토리지에 저장된 파일 다운로드하기 위하여 워크스페이스 컨트랙트를 통해 저장하고자 하는 파일의 액세스 키를 확인한다. 그 후, 분산 어플리케이션은 워크스페이스 컨트랙트를 통해 해당 파일에 대한 사용자 권한을 확인하고 앞서 설명한 업로드 과정과 유사한 과정을 통해 액세스 토큰을 전달받는다. 분산 어플리케이션은 전달받은 액세스 토큰을 사용자의 개인키를 통해 서명하고 액세스 토큰과 서명 정보를 REST API 서버의 파일 다운로드를 위한 웹 서비스를 요청을 통해 전달한다. REST API는 전달받은 액세스 토큰과 서명 정보를 통하여 사용자를 검증하고 웹 서비스 요청에 대한 응답으로 해당 파일을 전달한다.

BSWork에서의 오프 블록체인 스토리지에 저장된 파일의 삭제와 관련된 인증 과정은 워크스페이스 컨트랙트와 REST API 서버에서 2단계에 걸쳐 수행된다. 워크스페이스 컨트랙트는 사용자의 어카운트를 기반으로 관련 함수의 실행을 제어하며, 이를 위하여 함수 실행 가드인 Modifier를 통해 권한이 인증되지 않은 사용자의 어카운트는 관련된 함수를 실행할 수 없도록 한다. 또한, REST API 서버는 별도의 사용자 인증 과정을 사용하여 외부에서 REST API 서버의 웹 서비스를 직접 요청하지 못하도록 한다. 그림 10은 파일 삭제를 위한 과정을 나타낸다.

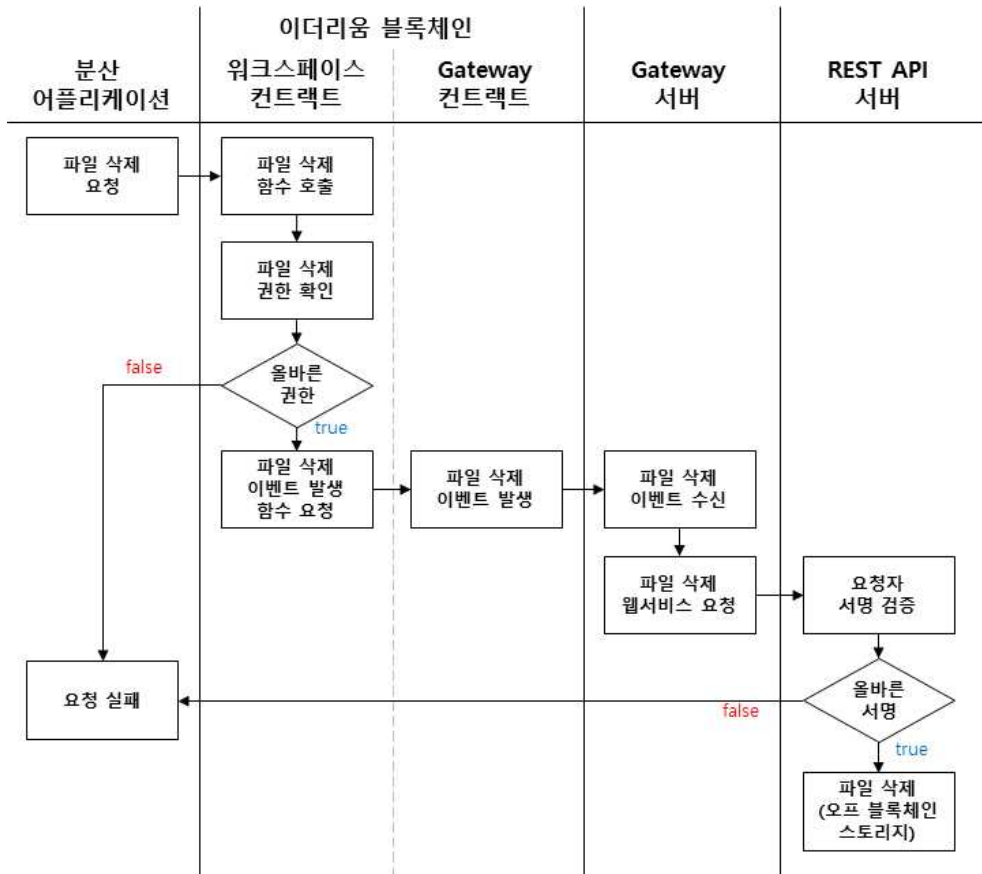


그림 11 오프 블록체인 스토리지 기반 파일 삭제 과정

분산 어플리케이션은 파일의 삭제 요청 시 해당 파일의 액세스 키와 액세스 키를 사용자의 개인키로 서명한 서명정보를 함께 전달한다. 워크스페이스 컨트랙트는 사용자의 어카운트가 해당 파일에 대한 올바른 권한이 존재하는지 확인하고 인증된 권한의 사용자이면 전달받은 서명정보와 액세스 키를 포함한 이벤트를 발생시켜 파일 삭제 웹서비스를 요청한다. REST API 서버는 전달받은 액세스 키와 서명 정보를 통하여 사용자의 서명을 검증하고 오프 블록체인 스토리지에서 해당 파일을 삭제한다.

## 4. BSWork의 구현

본 장에서는 3장에서 설명한 BSWork와 이를 위한 여러 기법들의 실질적인 구현에 대하여 설명한다. 또한, 구현된 BSWork의 유효성을 검증하기 위한 협업 참여자의 기여도 평가의 개발에 대하여 기술한다.

### 4.1 BSWork의 컨트랙트

BSWork는 이더리움 블록체인을 기반으로 기능을 제공하기 위하여 이더리움 스마트 컨트랙트를 이용한다. 본 절에서는 BSWork를 위한 이더리움 스마트 컨트랙트의 구조 및 주요 함수에 대하여 설명한다.

구현된 BSWork는 이더리움 스마트 컨트랙트를 통해 사용자 및 협업 그룹의 정보 저장 및 관리 등을 지원하고, 각 협업 그룹별 컨트랙트를 통해 그룹 구성원 및 협업 데이터를 관리한다. BSWork의 스마트 컨트랙트는 사용자 및 그룹 관리를 위한 컨트랙트인 MemberStore, GroupStore, WSFactory와 각 협업 그룹의 관리를 위한 컨트랙트인 UserGroup, GroupWorkspace 등으로 구성된다. 그림 11은 BSWork의 스마트 컨트랙트 구조를 나타낸다.

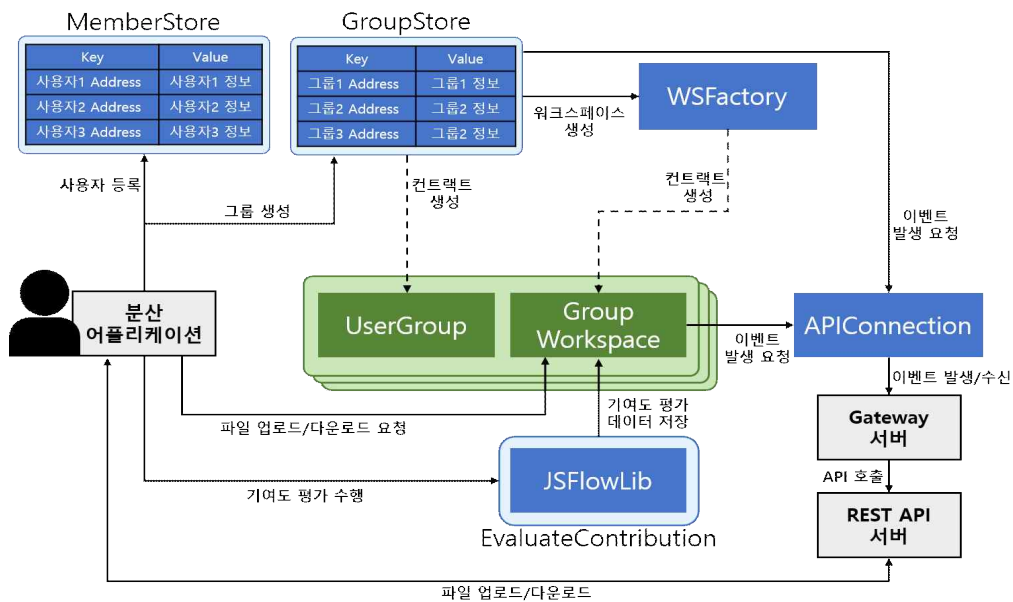


그림 12 BSWork의 컨트랙트 구조

MemberStore 컨트랙트는 BSWork 사용자의 관리를 위한 컨트랙트로서 사용자의 등록 및 정보 저장, 조회 등의 기능을 위한 함수를 제공한다. BSWork의 사용자는 MemberStore 컨트랙트의 newMember() 함수를 통해 자신의 정보를 등록할 수 있으

며, 등록이 완료된 사용자는 GroupStore 컨트랙트의 createGroup() 함수를 사용하여 자신의 협업 그룹을 생성할 수 있다. GroupStore 컨트랙트의 createGroup() 함수는 새로운 협업 그룹 생성 시 해당 그룹에 대한 UserGroup 컨트랙트를 생성하고 WsFactory 컨트랙트의 createGroupWorkspace() 함수를 통해 해당 그룹의 워크스페이스를 위한 컨트랙트를 생성한 후 이에 대한 정보를 저장한다. WsFactory 컨트랙트의 createGroupWorkspace() 함수는 GroupStore 컨트랙트에 의해서만 호출되며, 그룹의 워크스페이스에 대한 GroupWorkspace 컨트랙트를 생성한 후 생성된 컨트랙트의 정보를 반환한다. 표 1은 사용자 및 그룹 관련 컨트랙트의 주요 함수에 대하여 나타낸다.

표 1 사용자 및 그룹 관련 컨트랙트 및 주요 함수

컨트랙트	함수명	기능 설명
MemberStore	newMember	새로운 사용자 등록
	getMemberList	등록된 사용자 리스트 조회
	getMemberInfo	특정 사용자의 정보 조회
GroupStore	createGroup	새로운 그룹 생성
	getGroupList	생성된 그룹 리스트 조회
	getGroupInfo	특정 그룹의 정보 조회
	deleteGroup	그룹 삭제
WsFactory	createGroupWorkspace	새로운 그룹 워크스페이스 컨트랙트 생성

UserGroup 컨트랙트와 GroupWorkspace 컨트랙트는 협업 그룹별로 각각 하나씩 생성되며, 해당 그룹의 구성원 및 협업 데이터 관리를 위한 함수들을 제공한다. 각 그룹의 UserGroup 컨트랙트는 그룹명, 그룹장 정보, 그룹 멤버 리스트 등의 해당 그룹 및 그룹 구성원에 대한 정보를 저장하고, GroupWorkspace 컨트랙트는 해당 그룹에서 공유되는 파일들의 파일명, 사이즈, 해시값, 작성자 등의 파일에 대한 메타 정보들이 저장된다.

특정 협업 그룹에 가입하고자 하는 사용자는 UserGroup 컨트랙트의 requestApplicant() 함수를 통해 가입 신청을 하고, 그룹을 생성한 사용자인 그룹장은 acceptApplicant() 함수, rejectApplicant() 함수를 통해 해당 사용자의 그룹 가입을 승인 또는 거절할 수 있다. 또한, 그룹장은 inviteGroupMember() 함수를 통해 MemberStore 컨트랙트에 등록된 사용자를 자신의 그룹 멤버로 초대할 수 있고, ejectGroupMember() 함수를 통해 특정 그룹 멤버를 퇴출시킬 수 있다. GroupWorkspace 컨트랙트는 협업 그룹 구성원들 간의 협업 자원 및 결과물에 대한 저장 및 공유 등의 협업 데이



터 관리를 지원하기 위하여 디렉토리 생성, 파일 업로드 및 다운로드 등의 함수를 제공한다. 표 2는 협업 그룹과 관련된 컨트랙트 및 주요 함수에 대하여 나타낸다.

표 2 협업 그룹 관련 컨트랙트 및 주요 함수

컨트랙트	함수명	기능 설명
UserGroup	updateGroupInfo	그룹 정보 변경
	changeGroupOwner	그룹장 변경
	requestApplicant	그룹 가입 요청
	acceptApplicant	그룹 가입 승인
	rejectApplicant	그룹 가입 거절
	inviteGroupMember	그룹 멤버 초대
	ejectGroupMember	그룹 멤버 퇴출
	leaveGroup	그룹 탈퇴
	getGroupInfo	그룹 정보 조회
	getApplicantList	그룹 가입 신청자 리스트 조회
	getApplicantInfo	그룹 가입 신청자 정보 조회
	getMemberList	그룹 멤버 리스트 조회
	getMemberInfo	그룹 멤버 정보 조회
getWorkspaceAddr	그룹 워크스페이스 컨트랙트 주소 조회	
Group Workspace	createDir	새로운 디렉토리 생성
	uploadFile	새로운 파일 업로드 요청
	downloadFile	파일 다운로드 요청
	deleteFile	파일 삭제 요청
	getDirList	디렉토리 리스트 조회
	getFileList	파일 리스트 조회
	getFileInfo	파일 정보 조회
	returnAccessToken	파일 업로드/다운로드를 위한 액세스 토큰 전달

GroupWorkspace 컨트랙트의 uploadFile() 함수는 파일 업로드 시 호출되며 파일 크기나 파일 타입, 파일에 대한 해시값 등의 메타정보를 통해 해당 파일에 대한 키를 생성하고 생성된 키와 파일의 메타정보를 key-value 쌍으로 저장한다.

이와 더불어, BSWork의 스마트 컨트랙트에는 Gateway 서버와 연동하기 위한 API Connection 컨트랙트와 협업 참여자의 기여도 평가를 지원하는 EvaluateContributi

on 컨트랙트가 포함되며, 이에 대한 설명은 이후 4.3절과 4.5절에서 자세히 다룬다. BSWork의 컨트랙트들 중 MemberStore, GroupStore, WSFactory, APIConnection, EvaluateContribution 컨트랙트는 각각 하나씩 블록체인 네트워크에 미리 배포되어 사용되며, UserGroup, GroupWorkspace는 그룹 생성 시마다 동적으로 생성되어 배포된다.

## 4.2 JSFlow 라이브러리

본 절에서는 BSWork에서 JSFlow 명세를 기반으로 AND-OR 트리를 생성하고 이를 통한 작업 제어 기능을 제공하는 JSFlow 라이브러리의 개발에 대하여 설명한다. 개발된 라이브러리는 Solidity 라이브러리로 구현되어 JSFlow 명세를 통한 체계적인 작업 과정 수행을 지원한다. Solidity 라이브러리는 별도의 상태변수를 저장하지 않으며 다수의 컨트랙트에서 반복적인 재사용을 지원하는 Solidity의 기능이다. 그림 12는 BSWork에서의 JSFlow 라이브러리 활용 구조에 대하여 나타낸다.

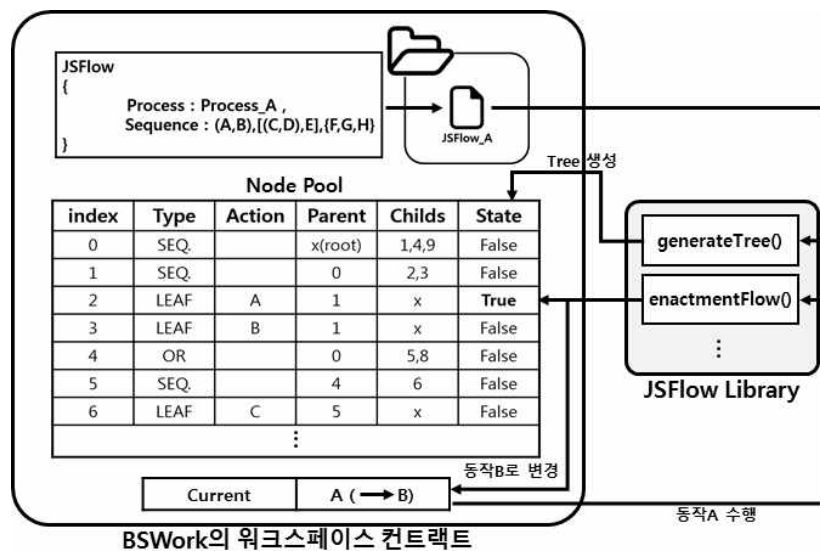


그림 13 BSWork에서의 JSFlow 라이브러리 활용 구조

JSFlow 라이브러리에는 AND-OR 트리와 트리의 노드 정보를 저장하기 위한 변수의 타입 등이 명시된 구조체인 TreeMap, Node에 대하여 정의되어 있다. 트리의 노드 정보에는 해당 노드의 색인번호(Index), 노드 종류(Type), 수행동작 정보(Action), 부모노드의 색인번호(Parent), 자식노드 색인번호 리스트(Childs), 동작수행 여부(State) 등이 포함된다. 또한, 트리의 정보에는 트리의 노드들을 저장하기 위한 배열 형태의 Node Pool과 현재 수행가능한 노드들에 정보가 저장된다.

각 노드의 색인번호는 Node Pool에서 특정 노드와 해당 노드와 연관된 부모노드

및 자식노드의 정보 조회를 위하여 사용된다. 또한, 노드의 수행동작 정보는 해당 노드가 수행가능 노드일 때 수행되어야 하는 동작에 대하여 기록되고, 동작수행 여부는 해당 노드의 동작이 수행 완료된 경우 True, 수행 전일 경우 False로 구분되어 기록된다. JSFlow 라이브러리를 사용하는 컨트랙트는 트리의 정보를 저장하기 위하여 라이브러리의 TreeMap 타입의 변수를 정의하여 가지고 있으며, 정의된 변수의 레퍼런스와 JSFlow 명세를 라이브러리에 전달하여 워크플로우를 지원하기 위한 기능을 사용한다. 개발된 라이브러리는 컨트랙트로부터 JSFlow 명세와 TreeMap의 레퍼런스를 전달받고, 이를 이용하여 AND-OR 트리를 생성한다. 또한, 생성된 트리를 기반으로 현재 수행가능 동작에 대한 정보 조회 및 동작 수행에 따른 상태 변경 등의 작업 흐름 제어를 지원한다. 이를 위한 JSFlow 라이브러리의 주요 함수는 표 3과 같다.

표 3 JSFlow 라이브러리의 주요 함수

함수 이름	내 용
generateTree	AND-OR 트리 생성
jsflowParser	JSFlow JSON로부터 데이터 구분 및 조회
initTree	트리 생성을 위한 초기화
addChildNode	특정 노드에 자식 노드 등록
setCurrentNode	현재 수행가능 노드 정보 변경
enactmentFlow	동작 수행에 따른 작업 흐름 진행
restartFlow	작업 흐름 초기화 및 재시작

BSWork의 워크스페이스 컨트랙트는 JSFlow 명세를 기반으로 트리를 생성하기 위하여 라이브러리의 generateTree() 함수를 호출한다. generateTree() 함수는 JSFlow 명세를 파라미터로 받아 initTree() 함수를 호출하여 트리를 초기화하고, jsflowParser() 함수를 통하여 JSFlow 명세로부터 작업흐름에 필요한 정보를 구분 및 조회한다. 또한, 조회된 정보와 addChildNode() 함수를 이용하여 트리에 노드를 추가하고, 트리 생성이 완료된 후 root 노드에서부터 탐색하여 현재 수행가능 노드에 대한 정보를 저장한다. enactmentFlow() 함수는 BSWork에서 현재 수행 가능한 노드의 동작이 수행되었을 경우 해당 노드의 동작이 수행되었음을 기록하고, 해당 노드의 부모노드를 조회하여 부모노드의 종류에 따라 다음 수행가능 노드를 탐색하여 이를 저장한다. restartFlow() 함수는 진행 과정 중 문제가 발생하거나 작업 흐름이 모두 완료되었을 경우 작업 흐름을 초기화하고 다시 시작하기 위하여 호출된다. JSFlow 라이브러리를 사용하기 위한 소스코드 예시는 그림 13과 같다.

```

import "../JSFlowLib.sol";

Contract JSFlowExample {
    using JSFlowLib for JSFlowLib.TreeMap;

    JSFlowLib.TreeMap exampleTree;
    ...

    function setJSFlow(string _jsFlow) ... {
        ...
        exampleTree.generateTree( _jsFlow ); // 트리 생성
    }
    function doAction_1() ... {
        ...
        exampleTree.enactmentFlow(/*Example Action 1*/); // 작업 흐름 진행
    }
    function restartFlow() ... {
        ...
        exampleTree.restartFlow( ); // 작업 흐름 재시작
    }
    ...
}

```

```

library JSFlowLib {
    enum NodeType{ SEQ, AND, OR, LEAF }
    struct Node {
        uint index;
        bool state;
        string action;
        NodeType type;
        uint[] childs;
        uint parent;
    }
    struct TreeMap {
        Node[] nodePool;
        uint[] currentNodes;
    }
    ... // 생략
}

```

그림 14 JSFlow 라이브러리 사용 예시

## 4.3 Gateway 서버

### 4.3.1 BSWork의 Gateway 컨트랙트

BSWork는 블록체인 네트워크와 오프 블록체인 스토리지간의 원활한 연동을 위하여 3.3절에서 설명한 Function Gateway를 기반으로 개발된 APIConnection 컨트랙트를 사용한다. 개발된 APIConnection 컨트랙트의 이벤트는 블록체인 네트워크에서 요청하고자 하는 REST API 서버의 API에 대한 정보를 포함한다. 이러한 이벤트들은 이를 발생시키기 위한 함수를 통해서 발생되고, Gateway 서버는 발생된 이벤트를 수신하여 해당 API를 요청한다. 표 4는 APIConnection 컨트랙트의 구조에 대하여 나타내고 있다.

표 4 APIConnection 컨트랙트 구조

```

contract APIConnection {
    event E_createGroup( address group, address workspace, address owner );
    ...
    function createGroup(address _gAddr, address _wsAddr) { //이벤트 발생 함수
        emit E_createGroup( _gAddr, _wsAddr, tx.origin ); // 이벤트 발생
    }
    ...
}

```

개발된 APIConnection 컨트랙트의 이벤트 발생 함수는 분산 어플리케이션이나 다른 컨트랙트로부터 호출되어 관련된 이벤트를 발생시킨다. APIConnection 컨트랙트에 정의된 이벤트에는 그룹이나 그룹 구성원의 정보가 변경됨을 알리기 위한 이벤트와 협업 데이터 파일 업로드 및 다운로드 등의 협업 데이터 파일에 대한 요청을 위한 이벤트 등이 포함된다. 표 5는 BSWork의 APIConnection 컨트랙트에 정의된 이벤트에 대하여 나타낸다.

표 5 APIConnection 컨트랙트의 이벤트

분류	이벤트명	파라미터	설명
Group	E_createGroup	그룹 컨트랙트 주소, 워크스페이스 컨트랙트 주소, 그룹장 주소	새로운 그룹 생성
	E_changeGroupOwner	그룹 컨트랙트 주소, 새 그룹장 주소, 기존 그룹장 주소	그룹장 변경
	E_deleteGroup	그룹 컨트랙트 주소, 그룹장 주소	그룹 삭제
Member	E_addMember	그룹 컨트랙트 주소, 멤버 주소, 그룹장 주소	그룹 멤버 등록
	E_deleteMember	그룹 컨트랙트 주소, 멤버 주소, 요청자 주소	그룹 멤버 삭제
File	E_uploadFile	그룹 컨트랙트 주소, 업로드 요청자 주소 워크스페이스 컨트랙트 주소, 파일 키, 파일 해시값, 파일 사이즈	파일 업로드 요청
	E_downloadFile	그룹 컨트랙트 주소, 다운로드 요청자 주소 워크스페이스 컨트랙트 주소, 파일 키	파일 다운로드 요청
	E_deleteFile	그룹 컨트랙트 주소, 워크스페이스 컨트랙트 주소, 파일 키, 삭제 요청자 주소, 요청자 서명	파일 삭제 요청
Email	E_sendEmail	이메일 주소, 이메일 데이터, 메일 타입	메일 발송

BSWork의 컨트랙트들은 새로운 그룹이 생성되거나 새로운 그룹 구성원이 등록되는 등의 그룹 및 그룹 구성원의 정보가 변경될 시 이를 REST API 서버에서 동기화할 수 있도록 APIConnection 컨트랙트에 정의된 이벤트를 발생시켜 관련된 정보를 전달한다.

#### 4.3.2 Gateway 서버 구현

BSWork의 Gateway 서버는 Node.js를[20] 기반으로 구현되었으며, APIConnection 컨트랙트에서 발생한 이벤트를 수신하여 해당 이벤트와 관련된 REST API 서버의 API 요청한다. 이를 위하여, APIConnection 컨트랙트의 코드를 Solidity 언어를 위

한 컴파일러 라이브러리인 Solc를[21] 통하여 APIConnection 컨트랙트의 ABI를 생성한다. 또한, 생성된 컨트랙트 ABI와 미리 배포된 APIConnection 컨트랙트의 주소 그리고 자바스크립트 기반의 Web3 라이브러리인 Web3.js를 통하여 컨트랙트 인스턴스를 생성한다. 구현된 Gateway 서버의 주요 코드는 표 6과 같다.

표 6 Gateway 서버 주요 코드

```

...
function genContract(_path) {                                //컨트랙트 ABI 생성
  let input = fs.readFileSync(_path, 'utf8');
  ...
  abi = output.contracts[contractName].APIConnection.abi;
  ...
  return abi;
}
...
Gateway = new web3.eth.Contract(                             //컨트랙트 Instance 생성
  genContract(GatewayPath), gatewayAddress
);
...
Gateway.events.allEvents({}, function (error, result) {     //이벤트 수신 대기
  switch (result.event) {
    case '이벤트 명':
      Request.post({                                         //API 요청
        headers: { 'Content-type': 'application/json' },
        url: '[API Server IP]:[Port]/[API 타입]/[API 명]',
        body: result.returnValues, json:true
      }, function (err, res, body) {
        ...
      })
      break;
  }
}
...
}
...

```

Gateway 서버는 APIConnection 컨트랙트에서 발생하는 모든 이벤트를 수신하고 수신한 이벤트의 이름을 통해 분류하여 관련된 API를 요청한다. 또한, 수신한 이벤트에 포함된 정보는 API 요청 시 JSON 형태로 요청 body에 포함되어 REST API 서버로 전달된다.

#### 4.4 오프 블록체인 스토리지 연동을 위한 REST API 서버

구현된 BSWork에서는 대용량 파일을 저장하기 위한 오프 블록체인 스토리지로 분산형 데이터베이스인 카산드라 DB를[22] 사용한다. 카산드라 DB는 데이터 관리의 안정성을 높이기 위하여 다수의 노드에 데이터를 분산 복제 저장하고, 이를 통해 데이터의 일관성을 보장한다. 본 절에서는 오프 블록체인 스토리지인 카산드라 DB와의 연동을 지원하기 위한 REST API 서버의 구현에 대하여 설명한다. 또한, 구현된 REST API 서버에서 사용자의 파일 접근 권한 인증 및 파일 검증 방법에 대하여 기술한다.

구현된 REST API 서버는 Node.js를 위한 프레임워크인 Express.js를[23] 기반으로 RESTful 웹서비스를 제공하고, Gateway 서버 및 분산 어플리케이션으로부터 전달받은 사용자의 요청과 연관된 동작을 수행한다. 또한, 카산드라 DB와의 상호작용을 위하여 Node.js 기반 카산드라DB 라이브러리인 Cassandra-driver를[24] 사용한다. 표 7은 REST API 서버에서 제공하는 웹서비스에 대하여 나타낸다.

표 7 REST API 서버의 웹서비스

분류	API	요청 body	응답	설명
/Group	/NewGroup	그룹 정보		새로운 그룹 생성
	/ChangeOwner	새 그룹장 정보		그룹장 변경
	/DeleteGroup	그룹 정보		그룹 삭제
/Member	/AddMember	멤버 정보		그룹 멤버 등록
	/DeleteMember	멤버 정보		그룹 멤버 삭제
/File	/ReqUpload	업로드 파일 정보	파일 업로드를 위한 액세스 토큰	파일 업로드 요청
	/ReqDownload	다운로드 파일 정보	파일 다운로드를 위한 액세스 토큰	파일 다운로드 요청
	/UploadFile	업로드 파일, 서명 액세스 토큰		파일 업로드
	/DownloadFile	서명, 액세스 토큰, 파일 정보	다운로드 파일	파일 다운로드
	/DeleteFile	서명, 액세스 토큰, 파일 정보		파일 삭제
/Email	/SendEmail	이메일 정보		메일 발송

블록체인 네트워크의 스마트 컨트랙트를 통해 새로운 그룹이 생성되거나 특정 그룹

에 새로운 멤버가 추가되면 이에 대한 정보는 APIConnection 컨트랙트와 Gateway 서버를 통해 REST API 서버로 전달되고 REST API 서버는 전달받은 정보를 카산드라 DB에 저장하여 블록체인 네트워크의 그룹 및 그룹 멤버 정보와 동기화한다. 동기화된 정보는 오프 블록체인 스토리지에 접근 시 해당 사용자의 그룹별 협업 자원에 대한 접근 권한 인증을 위해 사용된다. 그림 14는 REST API 서버에서의 파일 접근 권한 검증 구조를 나타낸다.

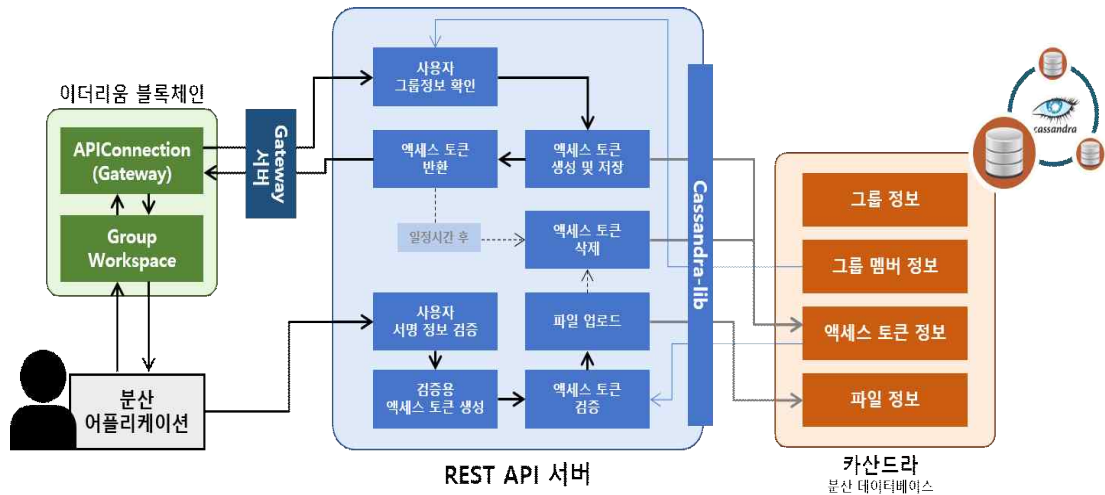


그림 15 REST API 서버에서의 파일 접근 권한 검증 구조

REST API 서버는 파일 업로드 요청을 수신하면 카산드라 DB에 저장된 그룹 및 그룹 멤버 정보를 통해 해당 요청자에 대한 권한을 확인한다. 이후, 액세스 토큰을 생성하여 이를 카산드라 DB에 저장하고 Gateway 서버와 블록체인 네트워크를 통해 사용자에게 반환한다. 또한, 해당 액세스 토큰을 카산드라DB에서 일정 시간 후 삭제하는 함수를 호출한다.

REST API 서버는 액세스 토큰이 삭제되기 전에 요청을 통해 파일, 액세스 토큰, 액세스 토큰에 대한 서명정보를 전달받게 되면 전달받은 서명정보 및 액세스 토큰과 Web3.js의 recover() 함수를 통하여 서명자의 주소를 확인하고 이를 카산드라DB에 저장된 액세스 토큰 정보와 비교한다. 또한, 확인된 서명자 주소와 전달받은 파일 정보 등을 이용하여 검증용 액세스 토큰을 생성하고 이를 기존의 액세스 토큰과 비교하여 검증 후 파일은 카산드라 DB에 저장한다.

#### 4.5 BSWork에서의 참여자 기여도 평가 서비스

본 논문에서는 구현된 BSWork의 응용 서비스로서 참여자의 기여도 평가 서비스를



개발하여 기능의 유효성을 확인한다. 이를 위하여 본 절에서는 BSWork에서 협업 참여자의 기여도를 평가하기 위한 서비스의 구현에 대하여 설명하고 이를 지원하는 기여도 평가를 위한 스마트 컨트랙트에 대하여 설명한다.

다수의 협업 환경에서 결과물에 대한 보상이나 권리 등이 협업 참여자의 기여도에 따라 분배되고 있다. 따라서 협업 참여자들의 기여도는 공정한 평가 과정을 통해 이루어져야 하며, 이러한 과정은 참여자들이 신뢰할 수 있어야 한다.

BSWork에서는 공정하고 신뢰할 수 있는 기여도 평가 과정 수행을 지원하기 위하여 JSFlow 명세와 JSFlow 라이브러리 이용한 스마트 컨트랙트 기반 기여도 평가 서비스를 제공한다. BSWork의 기여도 평가 서비스에서는 기여도 평가를 위한 EvaluateContribution 컨트랙트를 사용하며, 평가 과정을 JSFlow 명세를 통해 지정하고 이를 JSFlow 라이브러리를 통해 제어한다. 그림 15는 구현된 참여자 기여도 평가 서비스의 동작 구조를 나타내고 있다.

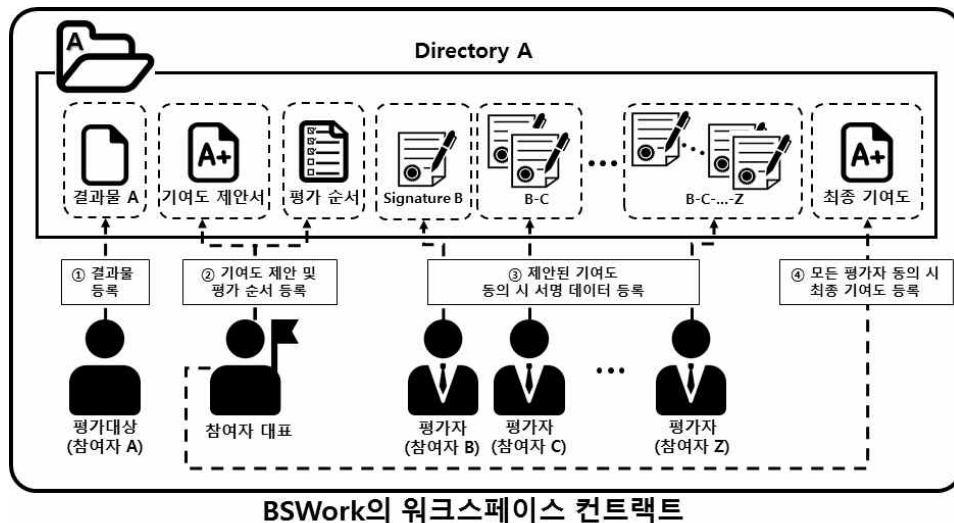


그림 16 BSWork에서의 기여도 평가 동작 구조

구현된 서비스에서는 평가 참여자들의 역할을 기여도 평가 대상인 “평가대상”, 평가대상의 기여도를 평가하는 “평가자”, 그룹의 그룹장인 “참여자 대표”로 구분하며, 평가대상과 참여자 대표는 그룹이나 평가 과정의 특성에 따라 평가자 역할 수행이 가능하다. 개발된 서비스의 기여도 평가 과정은 다음과 같다.

- (1) 참여자 대표는 EvaluateContribution 컨트랙트를 통해 평가 대상을 등록한다.
- (2) EvaluateContribution 컨트랙트는 협업 그룹 워크스페이스에 결과물을 저장하기 위한 평가대상의 디렉토리를 생성한다.
- (3) 평가대상은 협업 그룹 워크스페이스에 생성된 자신의 디렉토리에 결과물을 저장

한다.

- (4) EvaluateContribution 컨트랙트는 결과물이 저장되었음을 참여자 대표에게 메일을 통해 알린다.
- (5) 메일을 수신한 참여자 대표는 평가대상의 결과물을 검토하여 기여도 제안서와 이를 위한 평가 절차가 기록된 JSFlow 명세를 작성하여 평가대상의 디렉토리에 저장한다.
- (6) EvaluateContribution 컨트랙트는 저장된 JSFlow 명세를 기반으로 생성된 AND-OR 트리를 통해 현재 순서에 해당하는 평가자를 확인하고 이메일을 통해 평가를 요청한다.
- (7) 메일을 수신한 평가자는 평가대상의 결과물과 기여도 제안서를 확인하고 제안된 기여도에 대한 동의여부를 결정한다.
  - (7.1) 제안된 기여도에 동의할 경우, 평가대상의 결과물과 기여도 제안서 정보를 해시한 해시값을 자신의 어카운트를 통해 서명하고 서명데이터를 평가대상의 디렉토리에 저장한다.
    - (7.1.1) 이전에 평가를 수행한 평가자가 있을 경우, 이전 평가자의 서명데이터를 포함하여 해시한 해시값에 서명한다.
    - (7.2) 제안된 기여도에 반대할 경우, 명확한 반대 사유를 작성하고 이를 평가대상의 디렉토리에 저장한다.
- (8) 기여도 평가 컨트랙트는 가장 최근에 평가를 수행한 평가자의 평가 결과에 따라 다음의 과정을 수행한다.
  - (8.1) 평가자가 제안된 기여도에 동의할 경우 다음의 과정을 수행한다.
    - (8.1.1) 다음 순서의 평가자가 남아있을 경우, (6)의 과정을 수행한다.
    - (8.1.2) 모든 평가자의 평가가 완료된 경우, 평가 완료 상태를 참여자 대표에게 메일을 통해 알린다.
  - (8.2) 평가자가 제안된 기여도에 반대하였을 경우, 제안된 기여도가 반대되었음을 참여자 대표에게 메일을 통해 알리고 참여자 대표는 반대 사유를 확인한 뒤 (5)의 과정을 수행한다.
- (9) 참여자 대표는 저장된 기여도 제안서와 마지막 평가자의 서명데이터를 자신의 어카운트로 서명하여 최종 기여도 인증서를 작성하고 이를 평가대상의 디렉토리에 저장한다.

개발된 서비스를 통한 기여도 평가를 수행하기 위한 JSFlow 명세의 예시는 표 8과 같다.

표 8 기여도 평가를 위한 JSFlow 명세 예시

프로세스	JSFlow 명세
기여도 평가 과정	기여도 평가 과정 { <b>Process</b> : 기여도 평가 과정, <b>Sequence</b> : ( 기여도 제안서 등록, 평가 절차 등록 ), [ 기여도 평가, ( 제안 거절, 거절 사유 등록, 기여도 평가 과정 재시작 ) ], ( 최종 기여도 등록 ) }
참여자 A 기여도 평가	참여자 A 기여도 평가 { <b>Process</b> : 참여자 A 기여도 평가 , <b>Sequence</b> : ( 참여자 대표, 평가 대상A ), [ 평가자B, 평가자C ], [ 평가자D, 평가자E, 평가자F ] }

개발된 기여도 평가 서비스는 결과물에 대한 정보와 기여도 제안서, 이전 평가자의 서명정보를 포함한 해시값을 서명하는 체인형태의 서명방식을 통해 평가대상의 기여도와 평가 과정에 대한 위변조를 방지하여 신뢰성을 보장할 수 있다.

#### 4.6 분산 어플리케이션 동작화면

BSWork의 분산 어플리케이션은 웹 기반의 어플리케이션으로 개발되어, BSWork 사용자들이 웹 브라우저를 이용하여 각 기능을 수행할 수 있도록 인터페이스를 제공한다. 협업의 참여자들은 사용자 등록 화면을 통해 자신의 어카운트 정보를 저장할 월렛을 생성하고 생성된 월렛를 통해 BSWork의 사용자 등록이 가능하다. 그림 16은 사용자 등록을 수행하는 화면들을 나타낸다.



그림 17 사용자 등록 화면

협업의 참여자들은 BSWork의 사용자로 등록하기 위하여 자신의 프로필 정보를 작

성하고 작성된 프로필 정보를 MemberStore 컨트랙트의 newMember() 함수를 통해 등록한다. BSWork의 사용자 등록이 완료된 협업 참여자들은 자신의 그룹 리스트 화면에서 새로운 그룹을 생성할 수 있다. 그림 17은 그룹 생성 화면을 나타낸다.

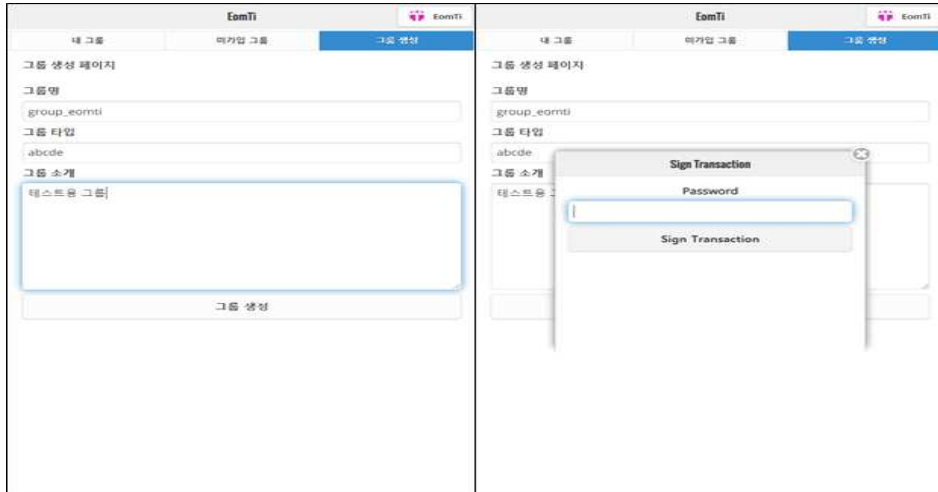


그림 18 그룹 생성 화면

BSWork의 사용자들은 생성된 그룹에 가입 신청 후 승인되거나 특정 그룹의 그룹장으로부터 초대를 받아 그룹의 워크스페이스에 접근할 수 있다. 이러한 그룹의 구성원들은 협업 워크스페이스 화면을 통해 자신의 협업 데이터를 업로드하여 서로간의 협업 데이터를 공유할 수 있다. 그림 18은 파일 업로드 화면을 나타낸다.

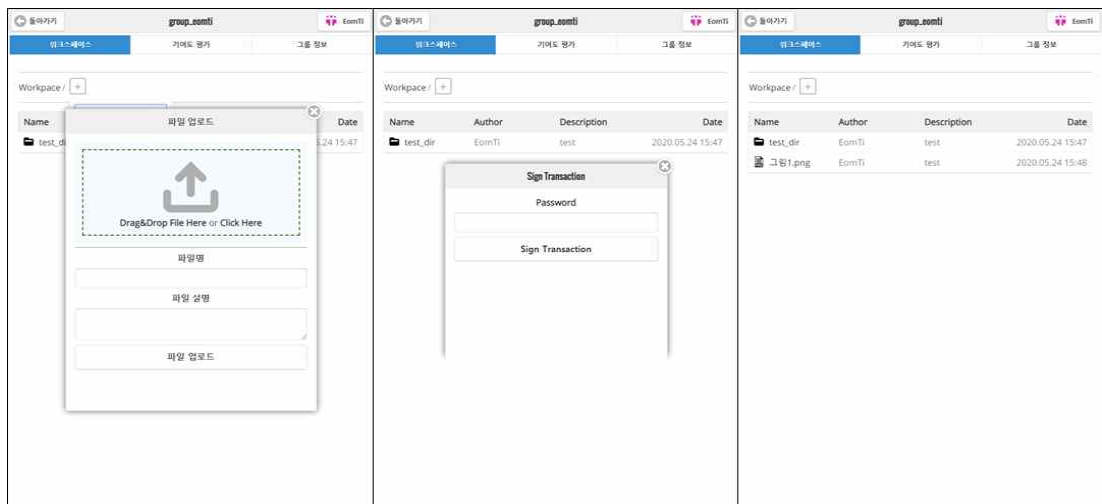


그림 19 파일 업로드 화면

협업 그룹의 그룹장은 기여도 평가 화면을 통하여 특정 구성원의 결과물에 대한 기여도 평가 과정을 등록할 수 있고, 협업 그룹의 구성원들은 기여도 평가 화면을 통하

여 해당 구성원의 결과물에 대하여 기여도를 평가하거나 평가된 기여도를 조회할 수 있다. 기여도 평가 화면은 그림 19와 같다.

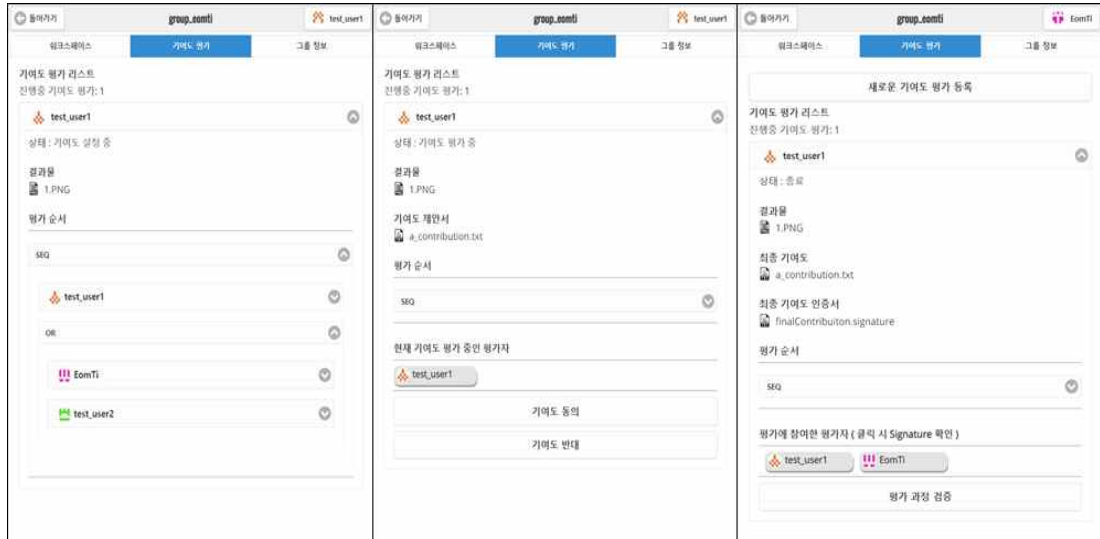


그림 20 기여도 평가 화면

#### 4.7 BSWork의 유용성 검증

본 논문에서는 실제 소규모 그룹들의 협업 수행을 위해 BSWork의 분산 어플리케이션을 사용하고, 해당 그룹들을 대상으로 설문조사를 실시하여 개발된 서비스의 유용성을 검증하였다. 이를 위하여, 실제 팀 프로젝트를 수행 중인 약 20여명의 대학생 및 대학원생을 대상으로 3주의 기간 동안 협업 수행을 위해 BSWork를 사용하도록 하였다. BSWork를 통하여 대상 그룹들이 수행한 협업 활동은 다음과 같다.

- 그룹 생성 및 그룹 구성원 추가
- 그룹 워크스페이스를 통한 구성원들 간의 자원 공유
- 결과물 저장 공간을 이용하여 자신의 결과물 등록
- 등록된 결과물에 대한 기여도 평가

사용자들은 BSWork를 이용하여 자신의 그룹을 생성하고, 협업 자원의 공유 및 결과물 등록 등을 통하여 협업을 수행하였다. 이후 사용자를 대상으로 실시한 설문조사를 통하여 해당 서비스에 대한 다양한 의견을 조사하였다. 그림 20은 실시한 설문조사의 결과를 보여준다.

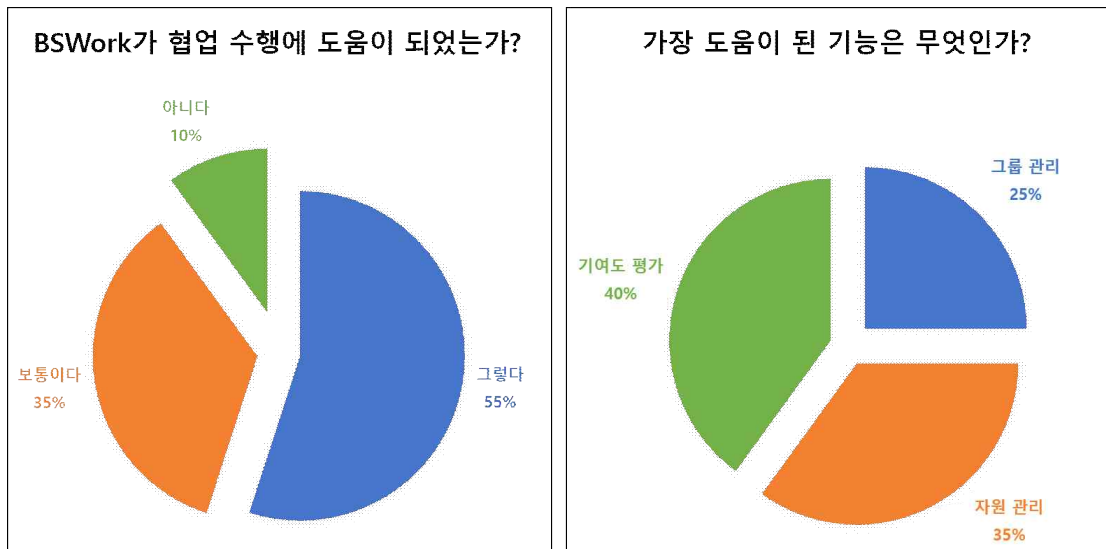


그림 21 설문조사 결과

설문조사 결과에 따르면 BSWork가 협업 수행에 도움이 된다고 답변한 사용자는 전체 사용자의 55%였으며, 유용한 기능에 대한 설문에는 25%가 그룹 관리, 35%가 자원 관리, 40%가 기여도 평가 기능을 선택하였다. 또한, 부정적인 답변을 한 사용자들은 해당 이유로 사용자 인터페이스 개선 필요, 응용 서비스의 부족 등을 언급하였다. 사용된 분산 어플리케이션이 기능 검증을 위해 프로토타입으로 개발된 어플리케이션임에도 불구하고, 과반수의 사용자가 긍정적인 답변을 보여 BSWork의 유용성이 검증되었다고 판단된다.

## 6. 결론

본 논문에서는 미리 지정된 작업 절차를 따라 신뢰성이 보장된 작업을 수행하도록 지원하는 블록체인 기반 협업 서비스인 BSWork의 개발에 대하여 기술하였다. 또한, 유동적인 작업 과정을 이더리움 스마트 컨트랙트에 간단하게 명세할 수 있는 JSFlow 기법을 제안하고 이를 BSWork에서 지원하기 위한 방법에 대하여 설명하였다. BSWork는 JSON 형태의 워크플로우에 대한 명세를 워크스페이스 컨트랙트에 저장하고 이를 통해 생성된 AND-OR 트리를 기반으로 작업 과정을 제어한다. 이를 위하여, 본 논문에서는 이더리움 스마트 컨트랙트에서의 JSFlow 명세를 통한 AND-OR 트리 생성을 지원하는 JSFlow 라이브러리를 개발하였다.

개발된 BSWork에서는 저장 가능한 데이터의 크기가 제한된 이더리움 스마트 컨트랙트의 한계를 보완하기 위하여 오프 블록체인 스토리지인 카산드라 DB에 협업 데이터 파일을 저장한다. 또한, BSWork의 REST API 서버와 Gateway 서버는 분산 어플리케이션과 카산드라 DB와의 연동을 지원하고 이에 대한 신뢰성을 보장한다. Gateway 서버는 분산 어플리케이션으로부터 스마트 컨트랙트에 요청된 그룹이나 파일과 관련된 사용자의 요청 및 정보를 Gateway 컨트랙트의 이벤트를 통해 수신하여 블록체인 외부의 REST API에 전달한다. REST API는 전달받은 정보들을 통해 블록체인 네트워크의 스마트 컨트랙트에 저장된 그룹 및 파일에 대한 정보를 동기화하며 이러한 정보와 액세스 토큰 등을 이용하여 접근 권한 제어 및 사용자 인증 문제를 해결하였다.

본 논문에서는 BSWork의 유효성을 확인하기 위하여 BSWork의 각 기능들을 활용하는 협업 참여자의 기여도 평가 서비스를 개발하였다. 개발된 서비스는 협업의 참여자의 기여도를 해당 참여자의 결과물을 통해 평가하며 이를 위한 평가 과정을 JSFlow 명세를 통해 제어한다. 또한, 협업 참여자의 결과물과 기여도 평가를 위한 다양한 데이터들은 BSWork의 카산드라 DB에 저장된다.

본 논문에서 개발된 블록체인 기반 협업 서비스는 기업이나 그룹간의 협업과 같이 협업 활동의 기록에 대한 신뢰성이 중요시 되는 작업환경에서 효과적으로 활용될 수 있을 것이다. 특히, 웹툰 작가 등의 다양한 창작 콘텐츠 제작자간의 협업과 같이 활동의 기록 및 결과물에 따른 기여도가 중요시되는 작업환경에서 협업 참여자들의 기여도를 보장하기 위한 방안으로 활용될 수 있을 것으로 기대된다. 또한, 본 논문에서 개발된 기법들은 협업 서비스뿐만 아니라 다양한 분야에서 분산 어플리케이션의 효율적인 개발을 위해 활용될 수 있다.

## 참 고 문 헌

- [1] Lee, Hong-Chang, Jong-Eun Park, and Myung-Joon Lee. "C3ware: A Middleware Supporting Collaborative Services over Cloud Storage." *The Computer Journal* 57.2 (2014): 217-224.
- [2] BSCW, <http://www.bscw.de/>
- [3] Alfresco, <http://www.alfresco.com/>
- [4] ROSARIO, JIMMY. "Technology BSCW (Basic Support for Cooperative Work) as Design Instructional for the Support to the Educational one, the Qualification and the Formation on the Basis of Manifold Theories of Learning." 2 nd International Conference on Information & Communication Technologies: from Theory to Applications, ICTTA. Vol. 6. 2006.
- [5] D. Georgakopoulos, M. Hornick, and A. Sheth. "An overview of workflow management: From process modeling to workflow automation infrastructure", *Distributed and parallel Databases* 3.2, (1995), pp. 119-153.
- [6] G. Zyskind, and O. Nathan, "Decentralizing privacy: Using blockchain to protect personal data", *Security and Privacy Workshops (SPW)*, IEEE, pp. 180-184, 2015.
- [7] H. Watanabe, S. Fujimura, A. Nakadaira, Y. Miyazaki, A. Akutsu, and J. Kishigami, "Blockchain contract: Securing a blockchain applied to smart contracts", 2016 IEEE International Conference on Consumer Electronics (ICCE), (2016), pp. 467-468.
- [8] V. Buterin, "A next-generation smart contract and decentralized application platform", *Ethereum project white paper*, (2014).
- [9] A. Bogner, M. Chanson and A. Meeuw, "A decentralised sharing app running a smart contract on the ethereum block-chain.", *Proceedings of the 6th International Conference on the Internet of Things*, (2016), pp. 177-178.



- [10] A. Bogner, M. Chanson and A. Meeuw, "A decentralised sharing app running a smart contract on the ethereum block-chain.", Proceedings of the 6th International Conference on the Internet of Things, pp. 177-178, 2016.
- [11] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system", (2008).
- [12] Solidity, <https://solidity.readthedocs.io/>
- [13] Web3.js, <https://web3js.readthedocs.io/>
- [14] J. H. Jin, H. M. Eom, J. E. Sun, and M. J. Lee, "A Blockchain-based Public Opinion Polling Solution Supporting Robust Verification", In: Proceedings of Convergence Research Letter (2017), Vol. 3, No. 3, pp. 245-248.
- [15] J. H. Jin, H. M. Eom, and M. J. Lee, "Bspace: a group workspace over the Ethereum blockchain with off-blockchain storage", International Journal of Advanced Computer Research(2019) Vol. 9, No. 40, pp. 53-60.
- [16] 김광훈, '워크플로우 기술 I', 한국정보통신기술협회(TTA) 저널, 85호, pp. 107-118, 2003
- [17] T. Hildebrandt, R. R. Mukkamala, and T. Slaats, "Designing a cross-organizational case management system using dynamic condition response graphs", 2011 IEEE 15th International Enterprise Distributed Object Computing Conference, (2011), pp. 161-170.
- [18] Y. Dong, Z. Shensheng, Modeling workflow process models with statechart, Proceedings of the 10th IEEE International Conference and Workshop on the Engineering of Computer-Based Systems (2003), pp. 55-61.
- [19] Yakindu, <https://www.itemis.com/en/yakindu/>
- [20] Node.js, <https://nodejs.org/>
- [21] Solc-js, <https://github.com/ethereum/solc-js>

- [22] A. Lakshman, and P. Malik, "Cassandra: a decentralized structured storage system." ACM SIGOPS Operating Systems Review (2010), Vol. 44, No. 2, pp. 35-40.
- [23] Express.js, <https://expressjs.com/>
- [24] Cassandra-driver, <https://github.com/datastax/nodejs-driver>

## ABSTRACT

### BSWork: A Blockchain-based Collaborative Service Supporting Workflow

Collaboration services provide functions such as creation and management of a collaboration group, data sharing between members, and project management to support smooth collaboration activities among collaboration participants. Currently, these collaboration services are used in various fields of work environment. In a collaborative environment in which many participants work in association with each other, work procedures are important and the participants must be able to trust these procedures and the results. So, it is desirable to support collaborative participants to draw results according to a designated work procedure, such as the sequence of individual tasks or the targets of performing tasks, and to provide a method for ensuring reliability.

In this thesis, we describe the development of BSWork, a blockchain-based collaboration service that supports reliably performing tasks through predefined work procedures. BSWork stores the specification of the workflow for the execution procedure of a collaborative work in the associated blockchain network and controls the process of the work through the specification. To this end, we present a simple method to specify workflows to collaboration groups of the blockchain and a technique to support the interaction with the outside of the blockchain network in BSWork. We also present an off-blockchain storage interworking technique to store and manage large amounts of data, such as resulting artifacts or collaborative resources derived through work procedures, while ensuring reliability.

**Key words** : Workflow, Collaborative Service, Blockchain, Ethereum, Decentralized Application