홍 승 필 교수 지도
석사학위 청구논문

디지털 비즈니스 환경 내 신뢰할 수 있는 블록체인 설계 방안

2017

성신여자대학교 대학원
컴퓨터학과
박 수 민

디지털 비즈니스 환경 내 신뢰할 수 있는 블록체인 설계 방안

홍 승 필 교수 지도

이 논문을 석사학위논문으로 제출함

2017년 5월

성신여자대학교 대학원

컴퓨터학과

박 수 민

인 준 서

박수민의 석사학위 논문으로 인준함

2017년 5월

심사위원장_____(인)

심 사 위 원_____(인)

심 사 위 원_____(인)

성신여자대학교 대학원

논 문 개 요

4차 산업혁명의 시대가 도래하며 빅데이터, IoT, 인공지능 등이 결합한 ICT 서비스가 각광받고 있다. 많은 데이터를 유통하고 저장하기 위해 블록체인이 4차 산업혁명의 요소기술로 주목을 받았다. 블록체인은 '신뢰할 수 있는 제 3자'가 없이 정보를 분산해서 관리하기 때문에 해커가 데이터를 해킹하기 어렵다는 이유로 보안성이 높은 기술로 각광 받고 있다. 특히 블록체인 기반의 스마트 계약은 P2P 환경에서 신뢰할 수 있는 계약 환경을 만들어주는 기술로 계약 코드를 통해 조건을 설정하고 계약을 작성하면 신뢰기관에 의존하지 않고 계약 당사자들 간에 정교한 계약을 실행할 수 있어 글로벌 기업에서도 주목하고 있는 기술이다.

하지만 블록체인 기술은 코드로 되어 있기 때문에 블록체인을 잘 이해하지 못한 일반 사람들은 스마트 계약을 활용하기에는 어려움이 존재한다. 그리고 블록체인은 다수가 같은 정보를 가지고 있기 때문에 무결성을 보장해주지만 데이터 보호를 위한 기밀성을 제공해주지 않기 때문에 개인정보나 민감정보가 보호되지 않고 블록체인에 업로드 되면 계약에 들어가는 사용자의 개인정보 및 민감정보를 유·노출에 대한 위협이 있다. 게다가 블록체인은 한번 블록이 생성되면 블록을 취소할 수 없기 때문에 블록체인에 데이터를 올리기 전에 비식별화 조치가 필수적이라고 사료된다.

본 연구에서는 신뢰할 수 있는 블록체인 설계 방안을 제시하여 안전한 환경에서 블록체인 기반의 스마트 계약 환경을 구축하고자 한다. 선행연구를 통해 블록체인 환경 내에서 발생할 수 있는 이슈들을 도출한 뒤 이를 토대로 블록체인 사용자 인증, 데이터 등급화를 통한 접근제어 기능과 쉽게 계약 코드를 파싱해주는 스마트 계약 컨트롤러의 기능을 가지고 있는 블록체인 설계 방안에 대해 제시한다. 제시한 설계 방안의 실 환경 적용 가능성을 검증하기 위해 블록체인 기반의 스마트 게임 머니를 사례연구와 설계 방안에 대한 향후 연구 방안에 대해 제시한 후 마친다.

목 차

표 차 례

그 림 차 례

제 1 장 서론

블록체인은 제4차 산업혁명의 요소기술로 기존에 중앙에 데이터를 저장하는 중앙집권식 서버와는 달리 네트워크에 연결되어있는 모든 사용자들이 같은 데이터베이스를 가지고 있는 분산 장부 시스템이다. 한국은행에서는 블록체인은 블록체인을 “P2P 네트워크에 거래정보를 기록한 장부를 분산해 관리하는 기술”이라고 정의했다[1]. 초기 블록체인은 처음 중앙은행 없이 화폐를 발행하는 비트코인(Bitcoin)의 저장 기술로 알려져 있었지만 블록체인의 뛰어난 보안성과 활용성에 주목하여 블록체인을 다양한 분야의 인프라로 사용하기 위한 연구가 주요국에서 활발하게 진행되고 있다. 특히 블록체인 기반의 스마트 계약은 블록체인 네트워크에 계약서를 올릴 수 있는 디지털 계약으로 제 3자가 필요하지 않고 컴퓨터 코드로 이루어져 있기 때문에 강제성을 지니고 있어 많은 대형 회사에서 블록체인 기반의 인증 프로세스는 물론 물류, 유통, 헬스케어, 부동산 등 다양한 인프라에 적용하기 위한 노력을 하고 있다. 다양한 분야에서 블록체인을 활용하는 사례가 늘어나고 있기 때문에 개인의 민감 정보, 식별정보에 대한 보호는 필수적이다.

하지만 블록체인 관련 활용 사례에 대한 연구는 활발하게 진행 중이지만 블록체인 내에 저장되는 개인정보와 민감정보를 보호하기 위한 연구는 미흡한 상황이다. 블록체인은 모든 데이터가 체인으로 연결이 되어 있어 데이터의 선택 삭제가 불가능하기 때문에 데이터의 관리가 중요하다. 이에 본 논문에서는 신뢰할 수 있는 블록체인 기반의 스마트 계약을 생성하기 위한 설계와 적용 방안을 제안하고자 한다.

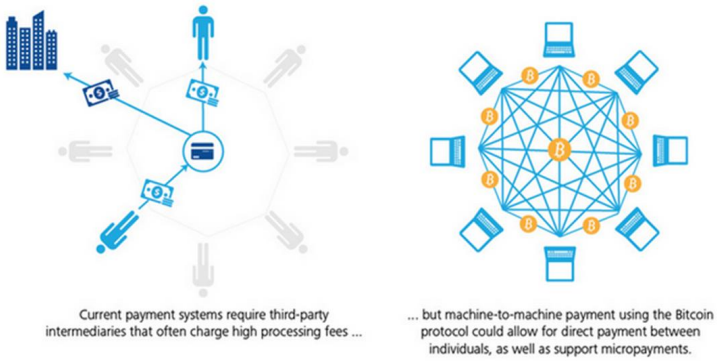
본 논문에서는 2장 블록체인 동향을 통해 블록체인의 정의와 동향에 대해 알아보고 3장에서 블록체인 환경 내 이슈를 도출하여 4장에서 이슈를 해결하기 위한 블록체인 설계 방안을 제안한다. 5장에서는 설계한 메커니즘의 알고리즘을 제시하고 활용 가능성을 검증하기 위해 블록체인 기반 스마트 계약을 활용한 스마트 게임 머니를 구현하고 6장에서 결론 및 향후 계획을 통해 마무리 한다.

제 2 장 관련연구

미국 가트너에서는 2017년 10대 전략 기술 트렌드 중 하나를 블록체인으로 선정했다. 블록체인은 비트코인 및 기타 토큰과 같은 가치 교환 거래가 블록 단위로 순차적으로 분류된 형태의 분산 장부이다. 각 블록은 기존 블록에 연결되고 P2P 네트워크를 통해 기록되며 암호화 트러스트 및 인증 방식을 사용한다. 블록체인은 비트코인으로 주목 받았지만 블록체인의 분산 장부 개념이 경영 모델에 적용할 수 있다는 가능성이 있다는 점에서 다양한 업계에서 활용할 수 있는 가능성을 보고 주목을 받았다. 2장에서는 블록체인의 정의와 동향, 특징 그리고 활용사례에 대해 서술한다.

1. 개요

한국은행에서는 블록체인을 '거래정보를 기록한 원장을 특정 기관의 중앙 서버가 아닌 P2P (peer-to-peer) 네트워크에 분산하여, 참가자가 공동으로 기록하고 관리하는 기술'로 정의하고 있다(한국은행, 2016).



(그림 1) 블록체인 개념도

출처 : Deloitte University Press, DUPress.com

블록체인은 기존 중앙 서버에서 거래를 승인해주어야 거래가 완료되는 중앙의 데이터베이스와 달리 블록체인은 P2P 네트워크 참여자들이 거래 승인을 통해 거래를 할 수 있는 구조이다. 블록체인 네트워크 내에 참여자들이 모두 같은 데이터를 저장하는 상호 분산원장을 이용하기 때문에 높은 보안성·확장성·투명성 등을 보장하는 기술이다.

블록체인 네트워크에 참여하고 있는 노드 간의 합의를 통해 블록을 업데이트하여 거래를 생성할 수 있다. 합의는 블록체인 내 데이터가 무결성을 가지고 있다는 것을 각 노드가 검증을 해주는 방식으로 블록체인 네트워크에 따라 다양한 합의 알고리즘을 가지고 있다.

[표 1] 블록체인 합의 알고리즘 종류

합의 알고리즘	주요 내용
PoW	- 거래 승인 과정에서 많은 컴퓨터 파워가 필요한 어려운 작업을 하는 노드가 신뢰할 수 있다고 판단
PoS	- 사용자의 소유 지분이 블록 생성의 우선권을 가짐
PoI	- 네트워크의 참여도를 통해 평가등급을 결정, 많은 양의 코인을 통해 거래를 하면 더 많은 보상을 가짐
Consensus-by -bet	- 참여자들의 동의를 통해 블록체인의 거래를 승인하는 방식

블록체인은 노드의 참여방식, 노드의 식별 가능성 등의 기준을 통해 공개(public) 블록체인, 컨소시엄(consortium) 블록체인, 사설(private) 블록체인 그리고 준 사설(semi-private) 블록체인으로 유형을 분류할 수 있다.

[표 2] 블록체인의 유형

<table>
<tr><th rowspan="2">분류</th><th colspan="3">비교 기준</th></tr>
<tr><th>참여 방식</th><th>노드의 식별 가능성</th><th>거래정보에 대한 접근</th></tr>
<tr><td>공개(public)
블록체인</td><td>누구나 참여
가능</td><td>식별 불가능</td><td>모든 참여자가 접근
가능</td></tr>
<tr><td>컨소시엄
(consortium)
블록체인</td><td>컨소시엄 멤버만
참여 가능</td><td rowspan="2">식별 가능</td><td rowspan="3">허가받은 사용자 및
거래 당사자만 접근
가능</td></tr>
<tr><td>사설(private)
블록체인</td><td>중앙기관 내
조직만 참여
가능</td></tr>
<tr><td>준 사설
(semi-private)
블록체인</td><td>정책에 따라
권한을 획득한
노드
(온라인으로
참여 가능)</td><td>식별 가능
(익명화 기능 제공)</td></tr>
</table>

※ 참고 : 금융보안원, 금융위원회

블록체인은 공개 블록체인, 컨소시엄 블록체인, 사설 블록체인, 준 사설 블록체인으로 분류 가능하며 참여 방식과 식별 가능성, 거래정보에 대한 접근 기준에 따라 나눌 수 있다. 대표적인 공개 블록체인은 비트코인, 이더리움 등이 있고 컨소시엄 블록체인과 사설 블록체인은 노드를 식별할 수 있다는 점에서 금융권, 사내 블록체인으로 활용한다. 준 사설 블록체인은 컨소시엄과 사설 블록체인과 달리 식별이 가능하지만 컨소시엄이나 사설 블록체인처럼 별명과 같은 ID로 식별이 가능하다는 점에서 국경 간 거래나 금융에서 유용하게 사용할 수 있다.

2. 블록체인 동향

1) 기술 동향

(1) 암호화 화폐

암호화화폐는 중앙의 기관보다 암호기술을 사용해 발행과 거래를 통제하는 새로운 형태의 통화라는 아이디어를 1998년 웨이 다이라는 개발자에 의해 제안되었다. 이 개념을 가지고 사토시 나카모토라는 익명의 개발자가 2009년 비트코인이라는 최초의 암호화화폐를 개발했다[2]

비트코인은 사토시 나카모토라는 익명의 개발자가 개발한 P2P 프로토콜로 화폐이지만 발행주체가 존재하지 않고 컴퓨팅 파워를 이용해 수학문제를 풀면 화폐를 채굴(mining)할 수 있는 방식으로 작업 증명 방식(Proof of Work : PoW)을 사용한다. 비트코인 이외에도 다양한 화폐가 합의 알고리즘이라는 것을 통해서 채굴이 된다.

(2) 스마트 계약

초기 블록체인은 암호화화폐를 저장하는 기술로 주목을 받았지만 블록체인을 플랫폼으로서 활용하기 위한 기술로 주목을 받은 것은 블록체인의 스마트 계약이다. 스마트 계약은 Nick Szabo에 의해 소개된 개념으로 신뢰할 수 있는 컴퓨터 인터넷 환경에서 “고도로 발달된” 계약을 준수하는 프로토콜로 정의할 수 있다[1)]. 스마트 계약은 블록체인에 코드를 업로드하여 실행하도록 하는 프로토콜로 비트코인, 이더리움 등에서 사용을 할 수 있다. 비트코인은 Script 함수를 사용하여 스마트 코드를 작성할 수 있다. 하지만 함수가 많지 않아 정교한 계약을 수행하기 힘들다는 특징을 가지고 있다. 이러한 개념을 확장하여 이더리움에서는 Vitalik Buterin은 튜링완전한 코드 개발을 통해 더 고도화된 형태의 스마트 코드를 개발할 수 있게 되었다.

1) Ethereum

이더리움 이외에도 R3CEV의 Corda나 IBM의 Hyperledger 플랫폼 역시 스마트 계약을 통해 다양한 종류의 스마트 계약을 사용할 수 있다.

스마트 계약은 소프트웨어에 의한 자동 실행이 가능하기 때문에 계약을 효율적으로 이행할 수 있게 되고, 제3자에 대한 의존성을 제거할 수 있어 비용 절감의 효과를 누릴 수 있기 때문에 다양한 분야에서 활용하기 위해 연구 중에 있다. 다음 표는 스마트 계약 활용 분야를 나타낸 자료이다.

[표 3] 블록체인 기반 스마트 계약 활용 분야

활용분야		내용
금융 서비스	송금 및 대출	- 신뢰할 수 있는 제3자 없이 송금 및 대출 가능
	보험	- 소액 보험금 계산 지급
헬스케어	전자의료기록	- 환자/의료공급자간 상호 승인에 근거한 전자의무기록에 대한 접근 허용 및 이전
	건강데이터 접근	- 의료연구자에 대한 개인의료 Data 제공
미디어	저작권 관리	- 자동으로 로열티(저작권료) 지급
공공 서비스	투표	- 투표자 자격검증, 투표기록 보관
	기록 관리	- 블록체인에 기록을 올려 관리
산업	공급망/무역금융서식	- 공금망을 따라 이동하는 상품 보관기록 증빙 - 상태변화에 따른 비용 지급
	P2P 거래	- 거래당사자 자동 매칭 및 대금결제

2) 플랫폼 동향

(1) 활용 분야

블록체인은 국내외 기관의 주요 트렌드에서 지능기술 관련이 핵심 이머징 이슈와 트렌드로 등장하였다. 가트너, OECD 그리고 세계경제포럼(WEF)는 블록체인을 10대 ICT 이슈 및 트렌드에 블록체인을 언급했다. 지난해 WEF는 “향후 1년 내 전세계 은행의 80%가 블록체인을 도입할 것”이라고 전망했다[2].

국내 기관인 한국정보화진흥원(NIA)에서도 정보보호의 중요성을 언급하며 생체인식, 블록체인을 차세대 보안 기술로 언급하여 블록체인에 대한 중요성을 강조했다. 또한 블록체인의 활용분야를 전자상거래, 스마트계약, 금융상품, 저작권 보호, 공공서비스, 사물인터넷으로 구분하여 다양한 분야에서 블록체인이 보안 플랫폼으로 활용될 수 있다는 것을 언급했다.

특히 IBM은 IoT, 해상 운송, 유통, 헬스케어 등 다양한 분야의 기업과 함께 블록체인 플랫폼을 개발하고 있다. Microsoft는 클라우드 기반 Azzure 위에 블록체인을를 연동하여 누구나 쉽게 블록체인을 활용할 수 있는 플랫폼을 개발하였다. 국내 기업 중 삼성SDS와 LG CNS 등 여러 기업에서도 자체 블록체인을 개발을 통해 신시장 개척을 진행 중에 있다.

다음 표는 블록체인 관련 개발 및 연구 중인 기업을 정리한 자료이다.

[표 4] 국내·외 기업의 블록체인 플랫폼 동향

주체		주요내용
글로벌	IBM	- (IoT 플랫폼) Ethereum 기반 IoT 플랫폼 개발 - (해상운송) 글로벌 1위 선사 머스크와 IBM이 협력해 블록체인 기술을 활용한 SCM 디지털화 솔루션 개발 - (기업용 블록체인) Hyperledger의 Fabric 기반한 첫 번째 블록체인 서비스인 IBM 블록체인 출시 - (식품 거래 블록체인) 미국의 대형마트인 월마트는 지난해 중국

주체		주요내용
		베이징에 '월마트 식품 안전 협력 센터'를 신설하면서 식료품을 운송·판매하는 과정을 추적하는 데 블록체인 기술을 도입함. - (헬스케어) 블록체인 기술을 이용해 의료 연구 및 기타 목적용으로 환자 데이터를 안전하게 공유하기 위함으로 EMR(Electronic Medical Record), 임상 시험, 게놈 데이터, 모바일 기기/웨어러블/사물인터넷(IoT)의 보건 데이터 등을 포함하여 다양한 출처의 환자 데이터를 연구할 계획
	Microsoft	- (클라우드 Blockchain 서비스) 자사 클라우드 서비스(Assure) 내 이더리움 기반 블록체인 서비스 시작 - 자·타사 클라우드 서비스 연동 모듈 개발 중
	Intel	- (자체 블록체인 플랫폼 개발) Intelledger 프로젝트 진행 중
국내	삼성SDS	- (기업용 블록체인) 금융뿐 아니라 타산업 영역에서도 범용적으로 사용할 수 있는 기업용 블록체인 플랫폼 Nexledger 개발 - (생체인증) 지문이나 음성 등 개인 생체정보 인증을 활용하여 단전하고 편리하게 데이터 접근이 가능한 Nexsign과 연동
	LG CNS	- 자본시장 등 거래시스템에 블록체인 결합 관련 연구 - P2P 장외주식 유통플랫폼 서비스인 'B-Trading'개발 등 거래의 완결성 확보하는 서비스 개발 중
	SK㈜ C&C	- (물류 플랫폼)물류 경로 추적 및 정보의 효율적 관리를 위한 '블록체인 물류 유통 서비스' 및 신용장(Letter of Credit)·선하증권(Bill of Lading) 등 국제 무역 필수 문서 대상 '블록체인 문서 전자화(Digital Asset)·인증 서비스' 개발 착수

블록체인의 금융 거래에서의 장점 때문에 많은 글로벌 은행과 국내 은행은 블록체인을 이용하여 금융 거래 플랫폼을 개발하고 있다. 대표적인 컨소시엄인 R3 CEV와 Hyperledger와 블록체인의 표준화를 연구하고 있는 W3C의 블록체인CG와 국내 여러 은행에서는 2016년 11월 은행권 블록체인 컨소시엄을 구성하고 블록체인을 이용한 금융서비스를 제공하고 있다. R3 CEV는 2017년 현재 글로벌 은행인 뱅크 오브 아메리카, UBS, 바클레이스

등을 포함한 50개가 넘는 글로벌 금융사들이 참여하는 블록체인 컨소시엄이다. 참여한 금융사가 아이디어를 제시하면 R3가 블록체인 기술을 개발하고 조사 활동을 돕는 활동을 하고 있다. 2016년 11월에 Corda는 블록체인을 기반으로 만든 분산 데이터베이스로 Corda를 통해 글로벌 원장을 구성하여 재무를 단순화하고 은행의 유지보수 비용을 줄이기 위해 개발되었다. 블록체인 CG는 금융권 외에 다양한 산업에 블록체인을 적용하기 위한 표준화를 연구하고 있다. Hyperledger는 리눅스 재단의 '범산업용 분산원장 표준화 프로젝트(cross-industy open standard for distrivuted ledgers)'를 개발하고 있다. 주로 기업결제, 상품추적 및 관리 등을 위한 산업용 공동 플랫폼으로 활용될 수 있을 거라고 예상하고 있다. 국내에서도 2016년 11월 16개 사원은행 및 2개 협력기관인 금융보안원과 금융결제원이 함께 '은행권 블록체인 컨소시엄'을 구성하여 국내에서도 블록체인을 도입하기 위한 노력이 계속되고 있다. 다음 표는 글로벌 블록체인 컨소시엄인 R3CEV와 Hyperledger, 블록체인 표준화 연구를 진행하고 있는 W3C와 국내 은행권 블록체인 컨소시엄에 대해 정리한 자료이다.

[표 5] 국내·외 블록체인 컨소시엄

컨소시엄 명	주요 내용
R3CEV	- 금융 분야에 특화된 Corda 플랫폼 개발 - 금융기관 간 계약을 기록, 관리, 동기화하기 위해 개발된 프라이빗 플랫폼 연구 컨소시엄
Hyperledger	- 블록체인 기반 플랫폼으로 기업에서 적용 가능한 표준적인 블록체인 플랫폼 기술 구현을 위한 목적으로 만들어진 컨소시엄 - Fabric 플랫폼은 스마트 계약을 이용한 블록체인 플랫폼
W3C	- 블록체인 관련 표준화 연구 단체
은행권 블록체인 컨소시엄	- 블록체인 기반의 고객인증 정보 공유 플랫폼 연구 - 향후 인증, 자금이체, 무역거래에도 활용하는 방안 검토 중

(2) 활용 사례

① R3CEV

Corda는 R3CEV에서 개발한 금융 분야에 특화된 블록체인 네트워크이다. 현재 글로벌 대형 은행 Corda 네트워크는 기존 공개(public) 블록체인과 다르게 거래에 관계가 있는 노드들에게만 공개되는 거래가 진행되기 때문에 준 사설(semi-private) 블록체인 네트워크를 사용한다. 준 사설 블록체인은 사설(private) 블록체인처럼 허가된 노드만 블록체인 네트워크에 참여할 수 있지만 공개 블록체인과 같이 노드를 식별할 수 있다. 하지만 준 사설 블록체인에서는 주소가 아닌 노드가 설정한 '별명'을 사용하기 때문에 익명화를 할 수 있다. 또한 사설(private) 블록체인처럼 허가된 노드만 블록체인 네트워크에 참여할 수 있다[3].

[표 6] Corda의 주요 특징

주요 특징	설명
네트워크 구성	- 준 사설 블록체인 형태로 네트워크를 구성하고 모든 노드는 TLS(Transport Layer Security) 기반의 암호화된 통신채널을 통해 직접 통신
합의방식	- 다양한 합의 알고리즘을 제공하는 합의 서비스를 제공
스마트 계약 기능 강화	- 다양한 형태의 비즈니스 계약을 정의 및 실행하기에 충분한 프로그래밍 명령어(JAVA)를 제공하고 노드 간에 다양한 포맷의 계약 정보(코드, 문서, 인증서 등)를 전송 가능하도록 지원
시스템 간의 통합 지원	- 각 금융 시스템의 데이터베이스로부터 대량의 데이터(거래정보 등)를 고속으로 가져올 수 있도록 네트워크 설계

출처 : 금융보안원

Corda의 대표적인 특징은 거래에 대한 정보를 네트워크에 포함되어있는 참가자들과 공유하는 것이 아닌 거래에 관계있는 사람들만 정보를 나눠 갖기 때문에 기존 블록체인에 비해 데이터에 대한 보호를 할 수 있다. 또한 거래를 할 때 공증인 노드가 존재하여 합의할 때 공증인을 선택하여 공증인이 세부적인 거래정보에 접근하여

선택적으로 합의를 하는 방식을 사용한다.

② Hyperledger

Hyperldeger는 블록체인과 관련된 기술에 대한 연구를 하는 프로젝트이다. Hyperledger에서는 Fabric, Iroha, Sawtooth Lake 등의 하위 프로젝트를 통해 블록체인 플랫폼에 대해 연구를 하고 있다. 그 중 Fabric은 블록체인의 엔진을 만드는 프로젝트로 블록체인의 핵심 기술인 합의 알고리즘에 대한 연결, 블록체인 내 사용자 인증 서비스나 스마트 계약 코드인 체인 코드 등을 적용하여 블록체인을 쉽게 접근할 수 있도록 한 블록체인이다.

Fabric은 대표적으로 멤버십 서비스, 블록체인 서비스, 체인코드 서비스로 분류할 수 있다. 멤버십 서비스는 사설 블록체인에서 사용자에 대한 인증을 통해 노드에 대한 신뢰성 확보와 데이터 접근 제어를 제공해주는 서비스이다. 블록체인 서비스는 다양한 합의 알고리즘을 통해 블록을 관리할 수 있게 도와주는 서비스이고 체인코드 서비스는 블록체인의 스마트 계약 코드를 실행할 수 있는 서비스로 Fabric은 SDK를 제공하여 쉽게 체인코드를 작성할 수 있도록 지원해주고 있다.

Fabirc은 최근 버전 1.0을 발표하고 기존 멤버십 블록체인에서 CA를 추가하여 노드에 인증을 통해 접근제어 메커니즘을 추가했다.

3) 법·제도 동향

블록체인이 다양한 분야에서의 활용 가능성을 보고 주요국에서는 블록체인 관련 법·제도와 규제에 대한 적용을 진행 중이다. 그 중 미국은 각 주마다 블록체인 관련 법안을 승인하여 활발하게 사용될 것으로 전망된다. 특히 Arizona 주에서는 스마트 계약의 합법적인 효력, 유효성, 집행 가능성이 있다고 판단하여 블록체인을 인증된 전자 서명으로 간주하는 법안을 승인하여 스마트 계약의 법적 효력이 생길 것으로

사료된다. 또한 New York주에서는 2015년 세계 최초로 비트코인 등 다양한 디지털통화 거래업체의 인가를 포함한 Bitlicense를 제정하였다. BitLicense는 암호화화폐의 지급·수취 등 모든 거래 관련 정보를 기록하고 보존하여 의심거래의 모니터링 및 감독 당국 보고의 의무화를 통해 블록체인의 암호화 화폐의 양성화를 위해 다양한 법·제도를 제정하고 있다.

미국 다음으로 가장 활발한 곳은 유럽이다. 유럽연합은 비트코인을 공식 화폐 통용을 승인하여 국내에서는 눈에 띄는 블록체인 관련 법·제도가 미흡하다.

[표 7] 블록체인 관련 법·제도 동향

국가	주요 내용
미국	- (Arizona 주) 스마트 계약의 합법적인 효력, 유효성, 집행 가능성이 있다고 판단하여 전자 서명으로 간주하는 법안 승인 - (New York 주) 2015년 세계 최초로 비트코인 등 디지털통화 거래업체 인가를 표함한 규제 법규 제정 - (BitLicense) 미국 뉴욕주의 BitLicense는 지급·수취 등 모든 거래관련 정보를 기록하고 보존하며, 의심거래의 모니터링 및 감독당국 보고를 의무화
유럽연합	- (유럽사법재판소) 비트코인 공식 화폐 통용 승인하여 부가가치세 면제 - (European Parliament) 유럽연합 집행위원회가 블록체인에 대해 '불간섭원칙(hands-off approach)'를 채택하도록 권고('Smart Regulation')
일본	- (자금결제법) 가산통화를 재산적 가치로 정의하고 자금세탁 방지, 결제 안정성 제고 등의 법안을 담아 자금결제법을 개정
중국	- (전자 화폐) 전자화폐 거래 중 일부는 은행 면허가 없다고 해도 대출 업무가 가능하도록 법 개정

3. 선행연구

본 논문에서는 블록체인을 활용하는 방안에 대해 제시하고자 한다. 따라서 선행연구로는 블록체인에 대한 기초연구와 플랫폼 개발을 진행한 응용연구를 검토하여 블록체인을 안전하게 사용할 수 있는 방안에 대해 모색한다.

이혁준, 이수미(2016)[4]은 블록체인 기반 화폐인 비트코인에서 발생할 수 있는 이중지불, 부정 인출의 문제점인 레이스 공격, 무작위 공격 등에 대해 분석하고 비트코인의 신뢰 구조와 이를 기반으로 발생할 수 있는 대응을 승인 횟수, 블록체인 모니터링, 네트워크 대응 등을 제시했으나 구체적인 대응 방안에 대한 설명은 미흡하다고 사료된다.

신다혜, 이종혁(2016)[5]은 비트코인에서 구현할 수 있는 스마트 계약과 이더리움에서 구현할 수 있는 스마트 계약을 비교하며 블록체인 2.0에 대해 소개하고 스마트 계약에서 발생할 수 있는 위험성인 작성자의 실수, 고려하지 못한 corner case, 익명성 등을 이슈로 발제하고 안전한 스마트 계약을 위한 보안 기법에 대해 연구를 하였으나 스마트 계약을 안전하게 사용할 수 있는 설계나 매커니즘에 관한 내용은 미흡하였다.

Ari Juels, Ahmed Kosba, Elaine Shi(2016)[6]은 스마트 계약이 은행 자동화 등의 이점은 있지만 제 3자가 없는 환경이기 때문에 범죄로 이용될 가능성이 있다고 말하며, 스마트 계약에서 발생할 수 있는 부작용에 대해 설명하고 이를 해결하기 위한 opcode 지원 및 인증된 데이터 피드 생태계에서는 기존 암호 기술로 효율적으로 구현할 수 있음을 알고리즘을 통해 증명했다. 하지만 블록체인 내에서 발생할 수 있는 범죄인 비밀 누출, 키 도난만 다루고 블록체인 거래 시 필요한 에스크로와 분쟁해결 등에 대한 연구는 미흡했다.

Christopher D. Clark, Vikram A. Bakshi, Lee Braine(2016)[7]은 스마트 계약 템플릿을 개발하여 스마트 법적 계약서에 대한 디자인과 포맷에 대한 연구를

제시하며, 메타 데이터, 디자인, 매개 변수에 대한 설계, 암호화 해싱 등 스마트 법적 계약 저장 및 전송을 위한 일련화된 형식의 디자인 환경을 연구했다. 하지만 본 연구는 금융에서의 법적 계약만을 다루고 있어 비금융에서의 계약을 활용하기에는 부족한 부분이 있었다.

유헌우(2016)[8]은 블록체인 기반의 전자투표 플랫폼을 연구하여 기존 온라인 전자투표 시스템의 문제를 해소하고 종이투표를 대체할 수 있는 새로운 투표 방식을 제안하였다. 기존 전자투표의 문제점이었던 해킹에 대한 위험을 블록체인을 활용해 안전한 플랫폼을 개발하였지만 유권자를 등록할 때 사용자 인증하는 부분이 블록체인 주소로 구성되어 있어 일반 유권자가 사용하기에 어려움이 있을 거라 사료된다.

문정환(2017)[9]은 오픈 마켓과 쿠폰 서비스 보안을 위해 블록체인 기반의 쿠폰 서비스 연구를 했다. 기존 쿠폰 서비스의 스미싱 등 다양한 보안 위협이 증가하는 이슈를 도출하고 쿠폰 조작 및 위변조를 해결하였다. 하지만 실제 쿠폰 서비스에 사용에 대한 연구가 미흡했다.

오성영, 이창훈(2017)[10]은 블록체인 기반의 부동산 시장에 대한 연구를 진행 했다. 최근 성장하는 부동산 시장에서 생길 수 있는 데이터 신뢰성 여부에 대한 이슈 도출하고 부동산 판매자, 구매자로 분류하여 신뢰할 수 있는 부동산 시장을 구축하고자 했다. 하지만 공증인이 없기 때문에 매물에 대한 신뢰도나 법적 타당성에 대한 출처가 분명하지 않아 실제 활용되기에는 미흡하다고 사료된다.

기존의 연구는 블록체인의 기본적인 특성인 P2P와 보안성에 초점을 맞춰 기존 서비스에서 보안 취약한 부분을 블록체인에 적용하여 새로운 서비스를 개발하는 연구가 대부분이었다. 하지만 다양한 서비스를 제공하기 위해서는 취소 불가능에 대한 이슈 해결하기 위한 연구나 P2P 네트워크에서 생길 수 있는 분쟁에 대한 고려에 대한 언급을 찾아볼 수 없었다. 또한 모든 연구가 코드베이스였기 때문에 일반인들은 실생활에서 활용하기 어렵다는 결론을 내렸다. 선행 연구를 통해 블록체인에 올라가는 데이터와 분쟁에 대한 기술적, 정책적인 연구 필요성을 찾을 수 있었다.

제 3 장 블록체인 설계에 대한 이슈

앞서 살펴본 바와 같이 블록체인은 중앙 서버가 존재하지 않는 구조로 이루어져 있어서 보안성을 가지고 있다. 이러한 이유로 많은 개인정보를 저장하는 금융권에서 데이터를 저장하고 유통하는 과정을 블록체인에 도입하려는 움직임이 있다. 하지만 블록체인은 기존 사용하던 데이터베이스 구조와 많이 다르기 때문에 블록체인을 금융권과 다양한 사업에 활용하기 위해 프라이빗 블록체인, 컨소시엄 블록체인 등 다양한 형태의 블록체인 네트워크가 개발 되었다. 3장에서는 신뢰할 수 있는 블록체인 환경을 위해 퍼블릭 블록체인 내 이슈를 분석하고자 한다.

[표 8] 블록체인 설계에 대한 이슈

이슈	주요 내용
데이터 노출 가능성	- 거래 내용이 모두에게 공개되어 개인정보, 민감정보 노출에 대한 이슈가 있음
취소 불가능	- 한번 승인이 완료되어 거래 장부에 올라가면 취소를 할 수가 없음 - 개인정보보호법의 파기에 관한 조항에 위배됨
일반 사용자의 접근이 힘듦	- 컴퓨터 코드로 작성되기 때문에 일반 사용자들이 활용하기 어려움
책임 소재의 불명확함	- 신뢰할 수 있는 제3자가 없는 P2P 거래에서 분쟁이 발생했을 때 책임질 대상이 불명확
관련 법/제도 및 규제의 부족	- 관련 법/제도가 미비하여 실제 환경에서 활용하기에 제약이 존재

퍼블릭 블록체인에서 스마트 계약을 실행할 경우 사용자에게 따로 사용자 인증을 진행하지 않기 때문에 노드의 검증이 이루어지지 않기 때문에 계약의 신뢰성을 확보하기 힘들다. 따라서 스마트 계약을 실행하고자 하는 노드에게는 공인 인증기관에 사용자 인증과 기기 인증을 받아 노드의 신뢰성을 확보해야 한다.

스마트 계약을 생성할 때 Solidity라는 언어로 작성을 해야 하기 때문에 코딩에

지식이 없는 일반 사용자가 계약을 작성하기 힘들기 때문에 활용성이 떨어 진다 . 또한 계약서를 작성할 때 법 관련 전문가가 존재하지 않는 다면 계약이 법적 근거를 매번 확인해야 하고 P2P 거래에서 유효성을 검증이 불투명하기 때문에 실생활에서 활용되기 힘들다는 이슈가 존재한다. 많은 사람들이 스마트 계약 서비스를 활용하기 위해서는 스마트 계약을 설계 해줄 수 있는 템플릿이나 계약 코드가 담긴 라이브러리나 법적 근거를 자동으로 확인해줄 수 있는 통합적인 서비스 제공하는 것이 필요할 것으로 보인다.

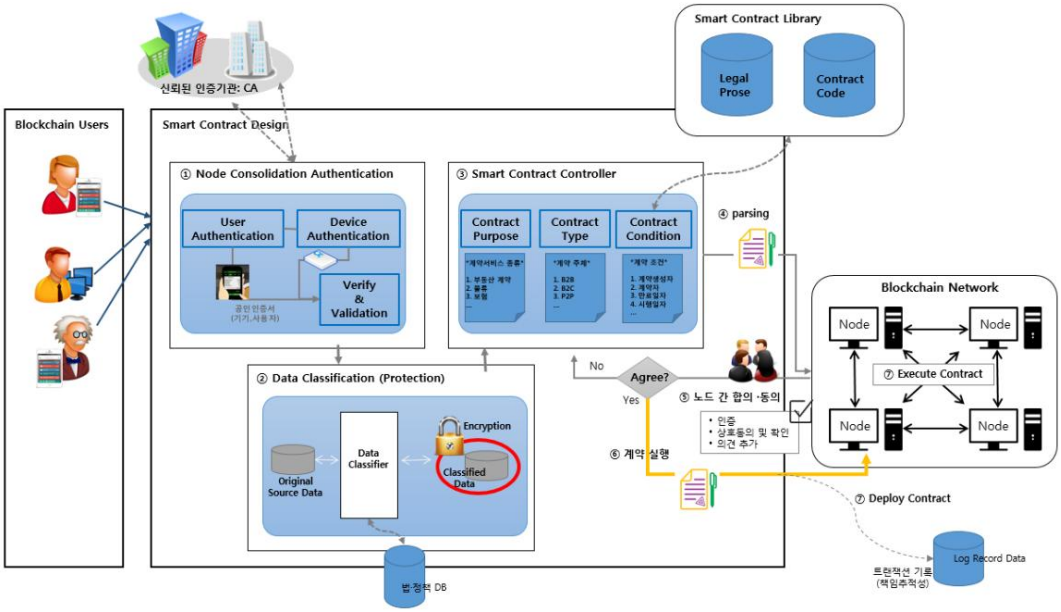
스마트 계약이 생성되어 블록체인에 올라갈 때 취소가 불가능하기 때문에 계약을 진행할 때 신중해야할 필요가 있다. 또한 계약 내에 쓰여 있는 계약자들의 민감 정보나 개인정보 등이 저장이 되면 개인정보보호법 상의 파기에 관한 이슈가 생긴다. 따라서 계약을 진행할 때 계약자의 개인정보를 저장하지 않거나 블록 안의 정보를 삭제할 수 있는 방안이 필요할 것으로 사료된다. 다음 장에서는 위에서 도출한 이슈를 토대로 안전한 스마트 계약에 대한 설계 방안 아키텍처를 제시하고자 한다.

제 4 장 신뢰할 수 있는 스마트 계약 설계 방안에 관한 연구

1. 전체 구성

본 논문은 블록체인 사용자들이 더 쉽고 신뢰할 수 있는 블록체인을 활용할 수 있도록 3장에서 도출한 이슈를 토대로 신뢰할 수 있는 블록체인 환경을 위한 스마트 계약 설계 방안에 대해 제안하고자 한다.

제안하는 설계 메커니즘은 아래 그림과 같이 나타낼 수 있다.



(그림 2) 신뢰할 수 있는 스마트 계약 설계 방안

본 연구에서는 블록체인에 대해 잘 알지 않는 사용자들도 신뢰할 수 있는 환경에서 블록체인을 활용하기 위해 위 그림과 같은 설계 방안에 대해 연구를 해보았다. 본

연구에서 제안하는 메커니즘은 크게 1) 사용자 통합 인증(Node Consolidation Authentication), 2) 개인정보보호를 위한 데이터 등급화(Data Classification), 3) 계약 목적, 유형, 조건등을 설정할 수 있는 스마트 계약 컨트롤러(Smart Contract Controller)로 구성된다.

먼저 블록체인에 접근하기 위해서는 블록체인 네트워크 사용자는 통합 사용자 인증을 받아 기존 퍼블릭 블록체인에서 노드를 식별 못해서 생기는 이슈를 해결하고자 한다. 사용자 인증은 본인 인증과 기기 인증을 받아 검증과 확인을 하여 인증이 된 노드만 블록체인 네트워크를 사용할 수 있도록 한다.

블록체인에 올라가는 계약 당사자들에 대한 정보는 데이터를 등급화 시켜 개인정보나 민감 정보를 보호한다. 원시 데이터를 데이터 분류기(data Classifier)를 통해 추출 된 데이터를 K-익명화 모델에 적합한 수준으로 비식별화 익명화를 시킨다. 이때 추출된 정보가 법적 적정성에 부합하는지 확인을 한 뒤 데이터를 분류한다. 분류된 데이터를 이용해 계약 조건을 적용한 뒤 블록체인 네트워크에 배치하여 계약을 실행할 수 있다.

계약을 생성하는 것을 도와주는 스마트 계약 컨트롤러를 이용해 계약의 목적, 계약의 형태 그리고 계약의 조건 등을 설정한다. 이때 설정한 계약의 조건이 현행법에 적용될 수 있는지를 확인해주는 법적 근거 DB에 접근해 검증한다. 또한 스마트 계약 코드가 담긴 라이브러리를 이용해 기존에 있는 계약 코드를 이용할 수 있도록 설정해 보았다. 본 논문에서는 이더리움 플랫폼으로 구현을 했기 때문에 EVM(Ethereum Virtual Machine)[2]에 올릴 수 있는 Solidity언어를 통해 스마트 계약을 생성했다. 스마트 계약 컨트롤러를 통해 정의한 계약서의 형태를 계약 코드로 변환하여 계약을 생성하여 계약의 계정을 생성한다. 계약을 할 노드들과 합의 및 동의를 통해 계약 조건을 생성할 수 있도록 한다. 이때 노드들은 설정한 조건에 대한 의견을 조정하고 수정을 하는 등 다양한 과정을 거쳐 계약의 조건을 합의한다. 이 때 조정이 되는 계약

2) Ethereum Virtual Machine : 튜링완전성을 지원하여 스마트 계약을 실행할 수 있는 코드(C++, Javascript, Python 등)를 작성하여 실행할 수 있는 소프트웨어

조건은 스마트 계약 컨트롤러의 템플릿을 통해서 조건을 수정할 수 있도록 하고, 조건이 수정될 때마다 블록체인에 기록이 되어 언제 수정이 되었는지 확인을 할 수 있도록 한다.

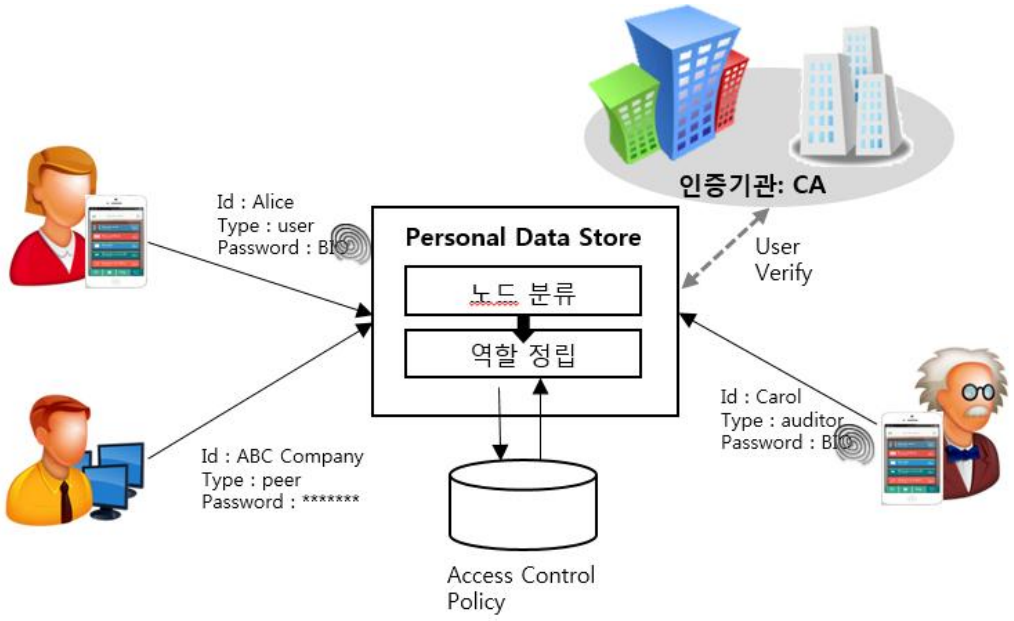
조건을 블록체인 네트워크에 배치하기 전에는 트랜잭션 로그 데이터베이스를 구축하여 계약 조건이 설정되어 배치 될 때 데이터베이스에 로그를 남길 수 있게 하여 추후 생길 수 있는 분쟁에 대비하여 근거자료로 활용할 수 있도록 한다. 다음은 세부 메커니즘을 자세하게 설명했다.

2. 세부 기능

1) 사용자 통합 인증

① 사용자 등록

신뢰할 수 있는 노드를 인증 받기 위해 사용자 통합 인증 프로세스를 제안한다. 신뢰할 수 있는 스마트 계약을 생성하기 위해 블록체인에 노드를 등록할 때 사용자 본인 인증은 물론, 기기 인증을 통해 1차적인 보안을 제공하여 안전한 블록체인 네트워크를 구성하도록 한다.



(그림 3) 사용자 등록

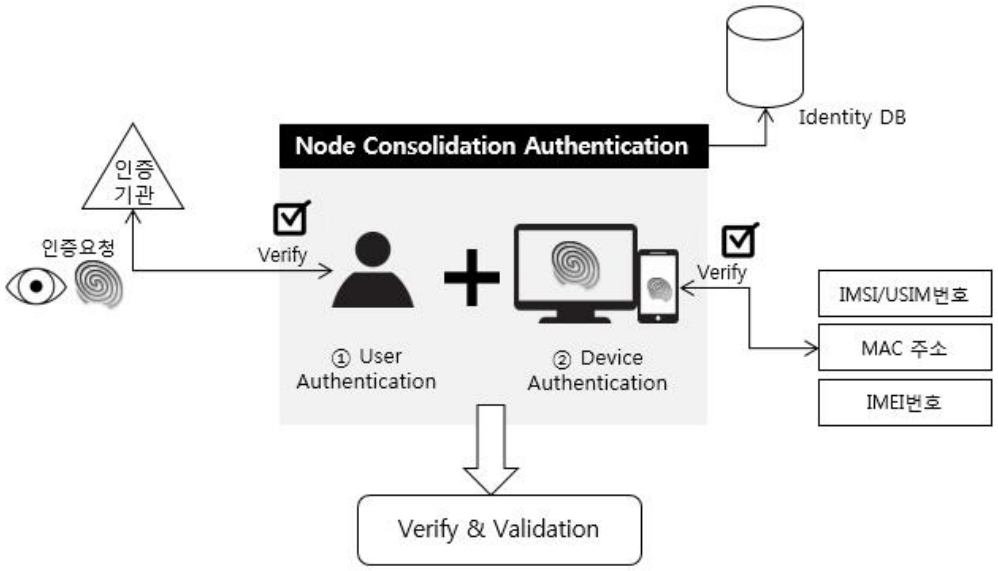
사용자 통합 인증은 사용자는 물론 사용하는 기기 인증을 받아 검증하고 확인하는 프로세스이다. 먼저 사용자는 블록체인 네트워크에 id와 Type을 설정한 뒤 생체 정보를 등록하여 가입을 한다. 본 논문에서는 user, peer, auditor로 구분하여 사용자의 등급을 분류해보았다.

[표 9] 블록체인 사용자의 분류

분류	세부 사항
user	- 블록체인 네트워크를 이용하는 일반 유저 - P2P, B2C 계약을 사용할 수 있음
peer	- 블록체인 네트워크를 이용하는 단체 유저 - 등록할 때 하위 레벨로 각 user를 등록
auditor	- 블록체인 내 거래를 감시하는 사람 - 모든 정보 확인 가능

id는 사용자가 편리하게 블록체인 네트워크에 접근할 수 있는 변수이고, Type은 블록체인 네트워크 내 노드의 등급을 분류하여 볼 수 있는 데이터를 접근제어를 위해 분류한다.

② 사용자 통합 인증



(그림 4) 사용자 통합 인증

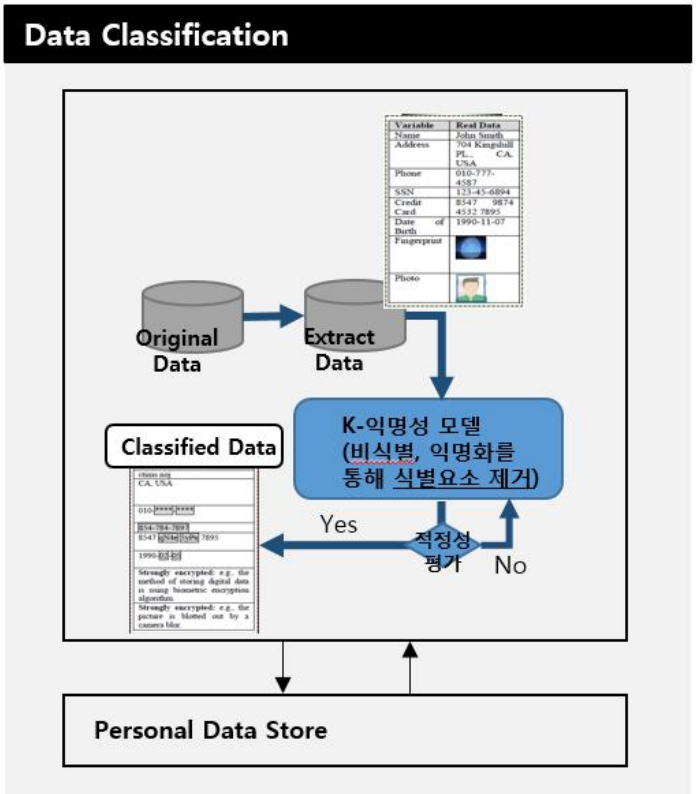
가입 후 사용자 인증은 FIDO(Fast IDentity Online) 기반의 지문, 홍채 등 생체 인식을 통해 검증을 하여 보안성을 높이는 것을 제안한다. FIDO는 기존의 비밀번호의 문제점을 해결하기 위해 제안된 사용자 인증 프레임워크이다[3]. FIDO 표준은 분실된 경우 스마트 기기의 안전한 영역에서 인증하기 때문에 사용자 생체정보가 외부로 노출되지 않아 안전하게 사용자 인증이 가능하다[11].

사용자 인증이 검증이 되면 기기 인증을 위해 사용자의 기기를 인증하기 위해 단말식별자인 IMSI/USIM 번호, MAC 주소, IMEI번호를 수집하여 최초 사용자 등록을 할 때 인증을 할 수 있도록 한다.

[표 10] 기기 인증 수집 내용

수집 내용	주요 내용
IMSI/USIM 번호	- 국제 단말 가입자 식별자로, 셀룰러 망의 사용자를 식별하기 위해 사용되며, 모든 셀룰러 망에서 유일한 식별자
MAC 주소	- 컴퓨터에 장착된 랜(LAN) 카드를 구별하기 위해 만들어진 식별 번호 - 공장에서 만들어져 나올 때 값을 설정하기 때문에 변경하기 어려움
IMEI번호	- 국제 모바일 단말 장비 식별자

2) 데이터 분류



(그림 5) 데이터 분류 메커니즘

사용자 통합 인증이 완료 되면 사용자의 개인정보 등을 추출하여 비식별화해 데이터를 보호하고자 한다. 본 논문에서는 개인정보 비식별화 가이드라인에서 제시한 k-익명성 모델에 맞춰 익명화를 제안한다. k-익명성 모델은 배포할 데이터 집합에서 준식별자 조합의 동일레코드를 k개 만큼 존재하게 하여 재식별 공격을 방어하는 비식별화 조치 중 하나다[12].

본 논문에서는 한국 CPO 포럼에서 제공하는 개인정보 영향도 등급분류[13]를 참고해 계약을 수행할 때 자주 사용할 것이라고 여겨지는 조합수준 P3, P2, S 등 개인정보와 서비스 이용정보 등을 비식별화 조치를 통해 블록체인에 저장하는 것을 제안한다.

[표 11] 개인정보 영향도 등급분류

조합수준	조합설명	개인정보영향도 설명
Privacy3	주민번호, 신용정보, 신용카드번호, 카드비밀번호, 계좌번호, ID/PW 등	개인의 신분 및 신상정보에 대해 알 수 있으며, 악용할 경우 위험이 매우 큰 정보
Privacy2	이름, 주소, 전화번호, 핸드폰번호, 이메일주소 등	개인의 신분과 신상정보에 대한 추정이 가능하여 노출 시 금액의 피해보상을 요구 받을 수 있는 수준
Privacy1	인종, 종교, 병역, 사회 단체활동, 보건 등	개인의 신분과 신상정보를 파악하기 어려우나 신상정보와 같이 노출 시 매우 민감한 정보
General	-	-
Service	상담내용, 녹취내용, 위치정보, IP정보, CCTV 영상정보, 카페이용내역 등	개인의 신분 및 신상정보에 대해 알 수 있으며, 악용할 경우 매우 큰 정보

출처 : CPO 포럼 개인정보 영향도 등급분류

분류를 한 데이터는 통합 노드 인증에서 분류한 노드의 역할에 따라 데이터 접근제어 서비스를 제공한다. 먼저 정보의 생성과 소비 주체를 정의하고 접근제어 권한 모델링을 통해 사용자 마다 접근할 수 있는 정보의 종류가 다르게 설정하는 것을 제안한다. 아래 표는 개인정보 영향도 등급분류를 토대로 노드에 따라 접근할 수 있는 정보와 데이터 등급을 A, B, C로 분류해 보았다.

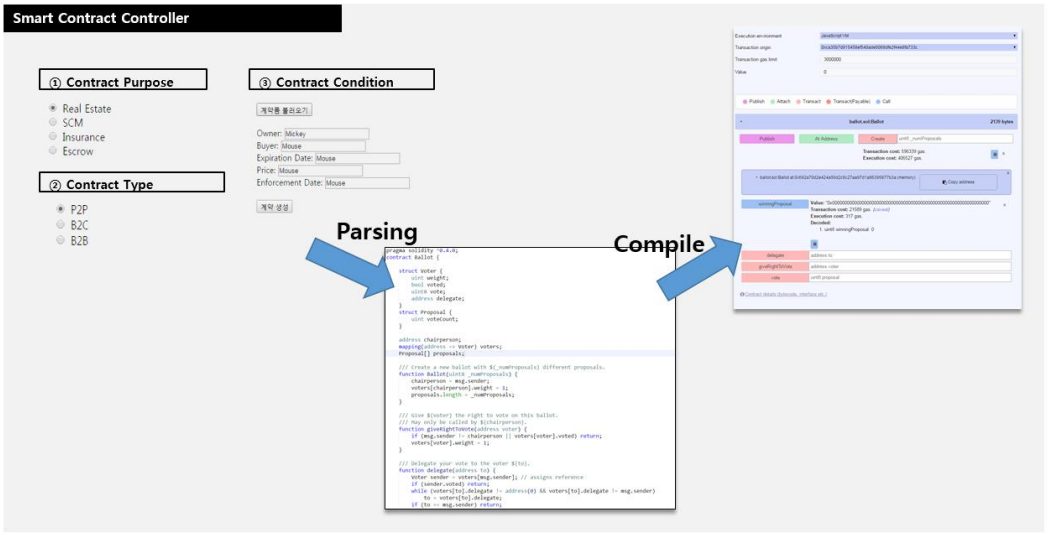
[표 12] 데이터 등급에 따른 접근 가능 노드

데이터 등급	데이터 정보	접근 가능 노드
A	- 주민번호, 신용정보, 금융정보 등의 개인정보 - 계약금 등과 같은 계약내용이 담긴 정보	auditor
B	- 상담내용, 녹취내용, 위치정보 등 서비스 이용정보	auditor, peer
C	- 일반 정보(나이, 성별, 통계자료 등)	auditor, peer, user

3) 스마트 계약 컨트롤러

스마트 계약 컨트롤러는 블록체인 네트워크 사용자가 코드 작성을 하지 않아도 쉽게 계약을 생성할 수 있게 도와주는 역할을 한다. 본 논문에서는 블록체인 네트워크에서 계약서를 작성할 때 1)계약의 목적, 2)계약 유형, 3)계약 조건을 설정할 수 있도록 설계했다.

본 논문에서는 이더리움 기반의 스마트 계약을 통해 설계를 했기 때문에 Solidity 언어로 계약코드를 작성하였다. Solidity는 스마트 계약을 실행할 수 있는 high-level 언어로 JavaScript와 유사한 형태를 가지고 있다. 상속, 라이브러리 및 다양한 기능을 지원하기 때문에 다양한 계약을 코드로 개발하는 것이 가능하다.

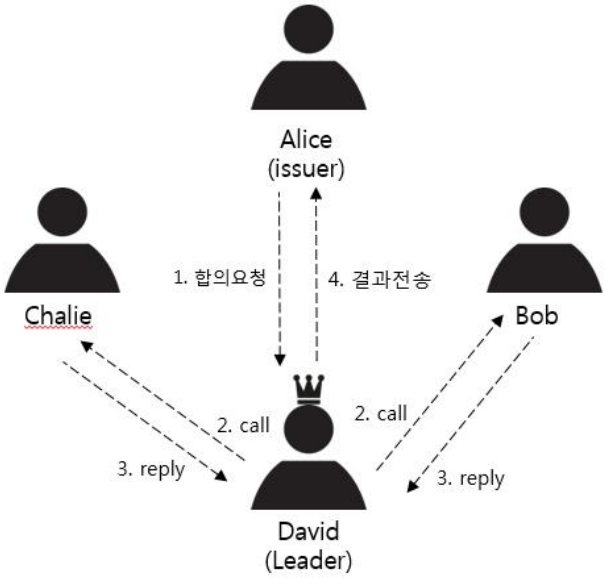


(그림 6) 스마트 계약 컨트롤러 프로세스(안)

계약 목적은 부동산, 유통, 보험 등 다양한 종류의 계약을 선택할 수 있다. 다음 계약의 유형은 개인 간 거래, 회사와 개인 간의 거래, 회사 간의 거래 등을 선택하여 다양한 유형의 계약을 생성할 수 있도록 설계했다. 마지막으로 계약 조건은 위

단계에서 설정한 계약의 목적, 계약 유형 등에 맞는 계약서 포맷이 생성되어 계약 당사자들은 해당 포맷에 정보만 채워 넣으면 되는 방식으로 계약서를 작성할 수 있다. 자주 사용하는 계약 내용은 EVM에 코드를 업로드 되어있는 정보를 데이터베이스에 저장하여 Smart Contract Library를 통해 제공을 하여 불러오기를 통해 스마트 계약을 생성할 수 있다.

계약 생성자가 설정한 계약 조건이 법적으로 타당한지 확인하기 위해 작성된 조건을 법률 언어로 파싱하여 타당성 여부에 대해 검사를 한다. 위 그림처럼 생성된 스마트 계약서는 법률언어(legal prose)를 스마트 계약 코드와 연결하여 계약을 실행하여 코드를 통해 계약의 유효성을 보증해주도록 한다.



(그림 7) 스마트 계약 컨트롤러의 합의 알고리즘

계약 생성 후 거래 참가자들과 합의를 통해 조건에 대한 수정을 진행한다. 본 설계 방안은 사설 블록체인을 기반으로 설계되었기 때문에 Raft 알고리즘을 사용하고 계약

검증은 auditor 노드를 통해 합의를 할 수 있도록 한다. Raft 알고리즘은 신뢰된 노드들로 구성된 사설 네트워크에서 진행하는 합의 방식으로 노드 중에서 임의의 대표 노드를 선출하여 대표 노드가 나머지 노드들의 동의를 얻어 합의를 하는 방식을 활용하는 것을 제안한다.

제 5 장 설계 및 프로토타이핑

1. 알고리즘

4장에서 제안한 신뢰할 수 있는 스마트 계약 활용 방안의 전반적인 기능을 알고리즘으로 제시하였다.

먼저 블록체인에 접근 하는 노드에 대한 사용자 본인 인증, 사용하는 기기의 정보를 수집하여 통합으로 인증하여 노드의 신뢰성을 확보한다. 다음으로, 노드가 계약을 생성하는 사람(issuer)인지 계약에 참가하는 사람(node)인지를 구분한 후 issuer일 경우에는 계약을 생성하고 계약의 조건을 우선 설정한 후 계약 조건이 법적으로 타당한지를 확인한다. 설정된 조건을 블록체인 네트워크에 전파하여 계약에 참여하는 참가자들 대상으로 합의 및 동의를 진행한다. 이때 과반수 합의가 되지 않을 때는 다시 계약 조건을 설정하여 합의가 될 때 까지 이 과정을 반복한다. 노드들 끼리 합의가 되었으면 계약에 들어가는 정보를 민감정보와 개인정보 등급을 나누어 비식별화 조치한다. 계약이 생성된 후에 생길 수 있는 분쟁에 대비하여 계약이 올라갈 때 마다 계약 데이터베이스에 저장하여 책임추적성을 제공할 수 있도록 한다.

아래 알고리즘은 설계 방안에 대한 알고리즘으로 실제 계약이 진행 될 때 노드의 분류는 더 세분화될 수 있다.

Algorithm

```
node ← node
node.user ← Blockchain User
node.peer ← Blockchain user group
node,auditor ←  Blockchain Network auditor

AccessControlPolicy DB ← 접근제어정책
Log DB ←  로그 기록

node.personalData ←  노드 인증 정보
```

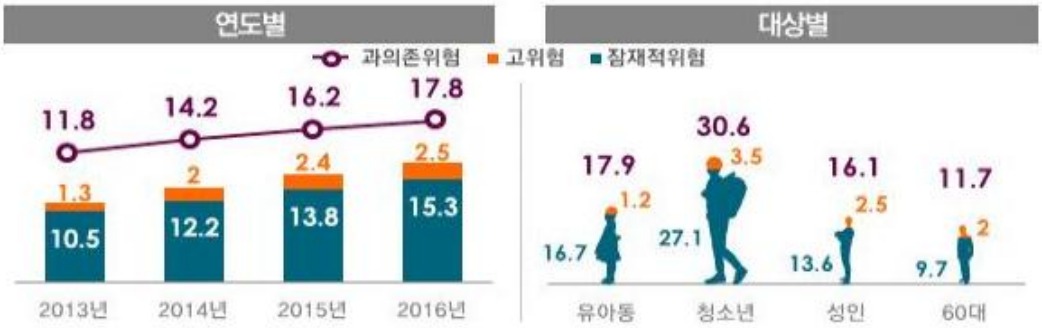
```
10:
11:   if isConsolidationAuth(node.id, node.type) = true then // 사용자(노드의) 통합 인증
12:   진행
13:     userAuthentication ← true
14:     nodeClassify(node.type) from AccessControlPolicy DB //노드 분류 및 역할 정립
15:
16:   else if
17:     userAuthentication ← false
18:     LogRecord(WARNING, authenticationFailMsg)
19:   end if
20:
21:     switch node
22:        case user :
23:             node ← user
24:        case peer :
25:             node ← peer
26:        case auditor :
27:             node ← auditor
28:
29:        while(node.personalData = k-model.suitable)
30:        {
31:           node.personalData ←  dataClassification(node.personalData)
32:        }
33:        contract.deploy
34:        LogRecord(TRANSACTION, transactionSuccessMsg)
35:        contract.execute //Execute Contract
36:
37:   if (node.issueContract) = true then // 계약 생성
38:     node ← issuer
39:     node.createContract
40:     contract.setConditon
41:     if contract.condition = law.suitable from legalProseDB then // 법적 타당성 검증
42:       while(allNodeAccept = true) // 노드 간 합의·동의
43:       {
44:          leader ←  electLeader(node) //Raft 합의 알고리즘을 위한 리더 선출
45:          issuer.requestConsensus then  //합의 요청
46:                leader.broadcastRequest then
47:          replyConsensusStatus // 합의 결과 전송
48:       }
49:
50:     end if
51:   end if
```

2. 프로토타이핑 : 스마트 게임 머니

본 논문에서는 위에서 제안한 설계의 타당성을 검증하기 위해 최근 이슈가 되고 있는 청소년 게임 문제를 해결할 수 있는 스마트 계약을 활용한 스마트 게임 머니를 구현해보았다. 먼저 게임 산업 내 이슈 중 청소년 게임 문제를 중심으로 도출하고 스마트 게임 머니의 법적 근거 확인한 후 설계 및 프로토타이핑을 통해 연구의 타당성을 검증하고자 한다.

1) 청소년 게임 문제

최근 청소년의 게임문제가 사회적 이슈로 대두되고 있다. 최근 연구 결과에 따르면 한국 청소년들은 일본 청소년보다 '인터넷 중독' 증상을 나타내는 비율이 4~5배에 이른다는 연구 결과가 나왔다[14]. 게임 중독이 심각하다. 게임 과외는 게임 속 신분이 현실 세계에서도 적용되기 때문에 게임에서 '패'하는 일이 없도록 고액 과외를 받는다고 설명한다. 이러한 문제 때문에 오전 0시부터 6시까지 만 16세 미만 청소년의 온라인 게임을 강제로 차단하는 '셧다운제'가 2011년부터 시행되고 있지만, 청소년들의 게임 중독 보호 효과를 제대로 가져오지 못하고 있다[15]. 또한 청소년들은 부모의 명의를 도용하여 선정적이거나 폭력적인 게임을 아무런 조치없이 이용하고 있다. 부모의 명의로 가입을 하여 사행성 게임을 하거나 아이템을 구입하여 게임을 이용하는 청소년도 적지 않다[16].



(그림 8) 연도별, 대상별 인터넷 과의존 실태조사

출처 : 미래창조과학부, 한국정보화진흥원. “2016 인터넷 과의존 실태조사 결과”

최근 청소년 관련 인터넷 등 중독 상담이 연 700여 건으로 게임에 중독된 사례가 적지 않다. 또한 부모의 휴대전화로 몰래 현금결제를 하여 200만원 넘게 과금이 되는 사례도 문제가 되었다[17].

청소년의 스마트폰 보급이 확산되면서 PC게임 이외에도 스마트폰 게임을 통해 시간과 장소에 관계없이 게임을 즐기는 청소년이 늘고 있어 게임에 대한 과 몰입이 사회적 문제로 확산되고 있다. 미래창조과학부와 한국정보화진흥원의 조사 결과[18]에 따르면 만 10~19세 청소년의 30.6%가 스마트폰으로 인한 금단, 내성, 일상생활 장애를 겪는 과의존(중독) 위험군으로 나타났다. 게다가 스마트폰을 통해 성인용 게임을 한 경험이 (32.2%)로 높게 나타났다.

본 논문에서는 위에서 도출한 대표적인 청소년 게임 문제인 게임중독, 부모 명의도용에 대한 사회적 이슈를 해결하기 위해 블록체인 기반의 스마트 게임 머니를 통해 구현하고자 한다.

2) 스마트 게임 머니 관련 법적 타당성 연구

스마트 게임 머니의 법적 타당성을 연구하기 위해 적용될 수 있는 법적 근거[19]에 대해 정리를 해보았다. 「청소년 보호법」의 인터넷게임 제공자의 고지 의무 항목을 살펴보면 16세 미만의 청소년이 게임을 이용할 때 관련 게임의 정보를 친권자에게

알려줄 의무가 있고 심야시간대의 인터넷 게임 제공시간 제한 항목에는 16세 미만의 청소년이 오전 0시부터 오전 6시까지 게임을 할 수 없는 항목이 존재한다. 따라서 본 논문에서는 부모가 게임 머니를 송금한 후 자녀가 게임 머니를 사용할 때마다 부모에게 게임 머니 사용 내역을 전송하도록 하고, 게임 머니 사용시간을 기본적으로 오전 0시부터 오전 6시까지로 설정할 수 있도록 개발하였다.

또 「개인정보보호법」의 개인정보의 파기에 대한 항목을 준수하기 위해 부모가 설정한 만료일자가 다되었을 때 코드를 통해 자동으로 데이터가 삭제될 수 있도록 함수를 사용하여 블록체인에서 데이터를 삭제하는 방안으로 설계하였다.

[표 13] 스마트 게임 머니 관련 법적 타당성 연구

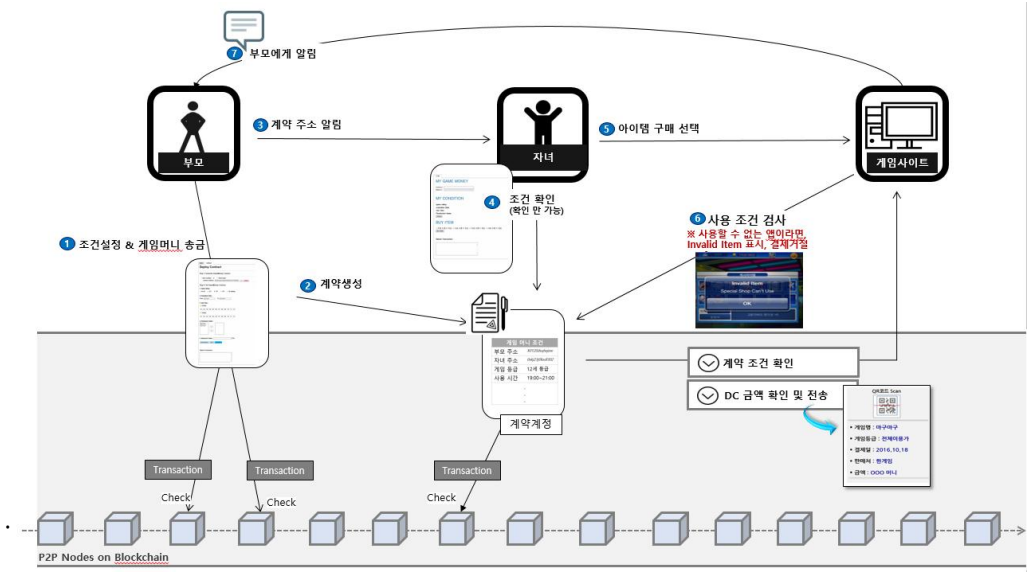
법 조항	주요 내용
「청소년 보호법」	- (인터넷게임 제공자의 고지 의무) 인터넷게임의 제공자는 16세 미만의 청소년 회원가입자의 친권자등에게 해당 청소년과 관련된 게임의 특성 및 등급, 인터넷게임 이용시간, 인터넷게임 이용 등에 따른 결제 정보 등을 알려야 함
	- (심야시간대의 인터넷게임 제공시간 제한) 인터넷게임의 제공자는 16세 미만의 청소년에게 오전 0시부터 오전 6시까지 인터넷게임을 제공하여서는 안됨
「개인정보보호법」	- (개인정보의 파기) 개인정보처리자는 보유기간의 경과, 개인정보의 처리 목적 달성 등 그 개인정보가 불필요하게 되었을 때에는 지체 없이 그 개인정보를 파기

3) 설계 및 프로토타이핑

(1) 설계

본 논문에서 연구한 스마트 게임 머니의 실제 활용 가능성에 대한 검증을 위해 테스트 환경을 구축하고 간략한 프로토타이핑을 구현하였다.

테스트 환경은 게임 머니를 송금하는 부모 지갑, 송금 받는 자녀 지갑으로 구분되며 각 지갑은 Windows 7의 64비트 환경에서 구현 하였으며 이더리움 플랫폼인 Go-Ethereum 1.4.18버전을 통해 개발하였다. 사용 언어는 Solidity를 통해 스마트 계약을 생성하고 지갑의 UI를 Javascript, HTML을 통해 사용자가 쉽게 사용할 수 있도록 계약서 포맷을 개발하였다. 또한 원활하게 게임 머니를 주고받기 위해 이더리움의 테스트 네트워크(test-net)에서 프로토타이핑을 수행하였다.



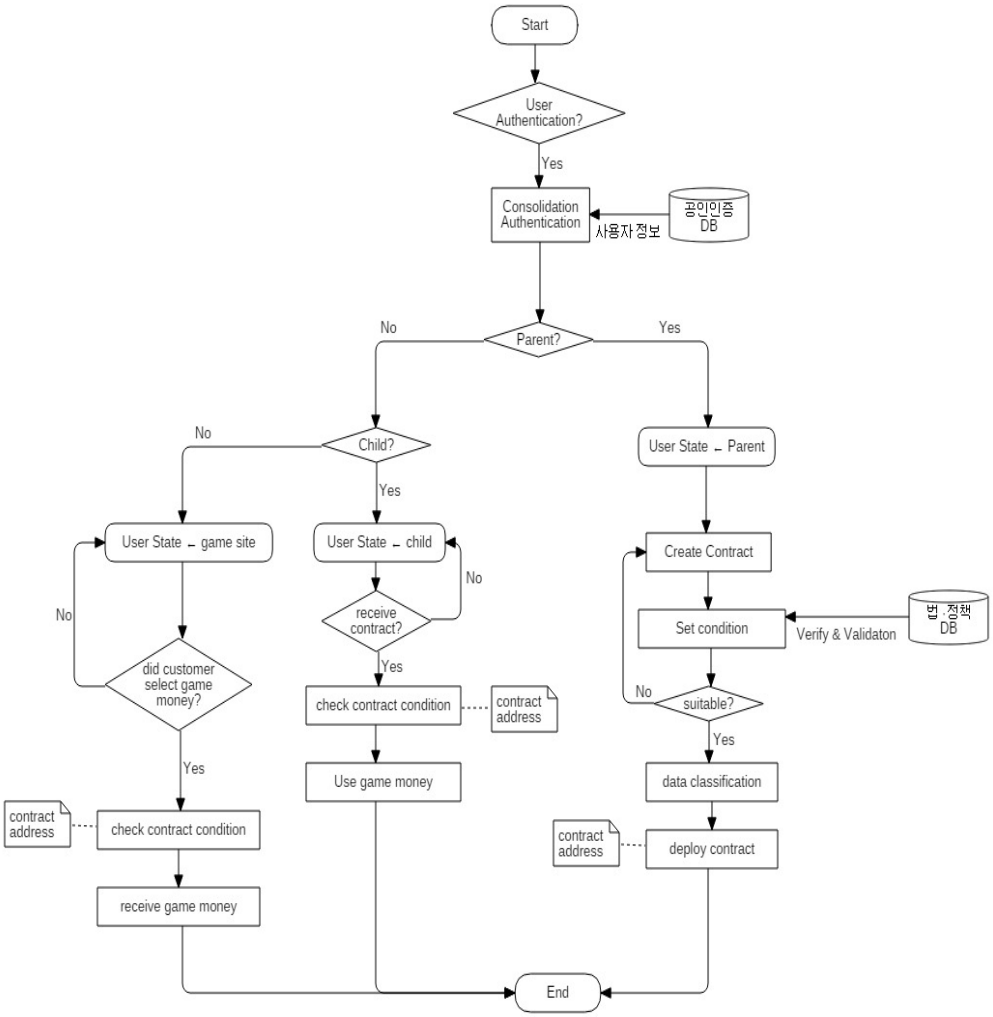
(그림 9) 스마트 게임 머니 시나리오

다음 표는 부모가 지갑을 통해 설정할 수 있는 게임 머니 조건에 대해 정리한 자료이다.

[표 14] 법적 근거를 통해 도출한 게임 머니 설정 조건

구분	세부 조건	설명	비고
게임등급	전체 이용가	- 게임등급을 확인하여 게임머니에서 허용된 등급의 게임에서만 사용 가능	필수
	12세 이용가		
	15세 이용가		
	청소년 이용불가		
	평가용		
유효 기간	직접 선택(게임머니 유효기간)	- 특정 기간 내에 미 사용시 회수	필수
사용 시간	셧다운제(24:00~06:00)	- 셧다운제 대상(만16세 미만)일 경우 특정 시간 동안 게임 사용 불가	선택 (기본값: 셧다운제)
	직접 설정(게임 가능 시간)	- 부모 등 게임머니를 구매하거나 전달해주는 사람이 사용 시간을 설정	

부모는 부모 지갑을 다운 받아 위 표와 같이 조건을 설정하여 자녀의 이더리움 지갑으로 게임 머니를 송금한다. 자녀는 부모에게 받은 계약 주소를 통해 자녀 지갑에서 게임 머니 사용 조건에 대해 확인을 할 수 있고 이더리움을 취급하는 게임 사이트에서 조건에 맞는 아이템을 게임 머니를 통해 구매할 수 있다. 자녀가 게임을 구매하면 게임 사이트에서는 부모에게 알림을 보내 자녀가 언제 어떤 아이템을 샀는지 확인할 수 있고 모든 거래 내역은 블록체인에 기록이 되기 때문에 자녀의 게임 사용 패턴에 대해 관리가 가능하도록 구현해보았다. 아래 그림은 위의 과정을 플로우차트로 나타낸 자료이다.



(그림 10) 스마트 게임머니 프로세스

(2) 프로토타이핑

스마트 게임 머니를 실행하기 위해 스마트 게임 머니 계약 코드의 일부를 제시한다. 아래 코드는 Solidity 0.4.2 버전으로 작성되었다. modifier를 통해 계약을 생성한 부모 사용자만 계약 조건을 수정할 수 있도록 onlyParent()와 게임 머니를 송금 받은 자녀만 사용할 수 있도록 onlyChild()를 선언하였다. 또한 kill() 함수를 통해 게임 머니 만료 기간 이후에는 계약서에 접근할 수 없도록 설정하여 개인정보보호법의 파기 항목에 위배되지 않도록 설정을 하였다.

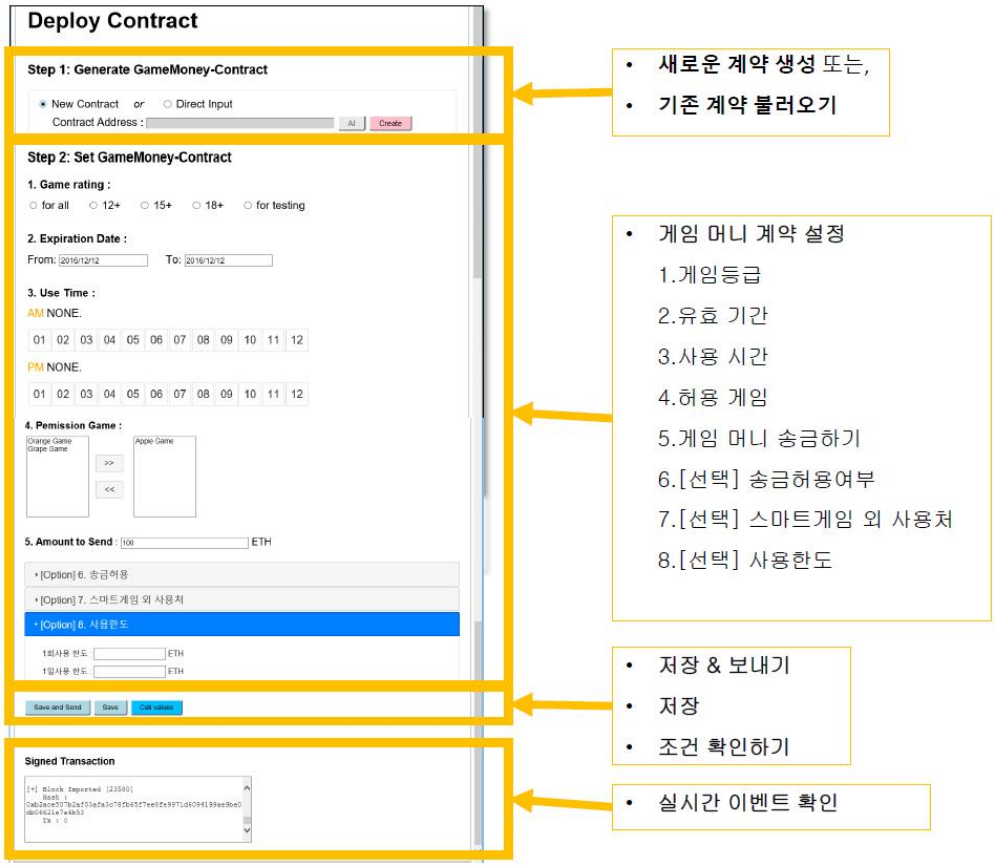
Contract GameMoney

```
1:  pragma solidity ^0.4.2;
2:  contract token { function transfer(address receiver, uint amount){  } }
3:  contract GameMoney{
4:    address public parent;
5:    address public child;
6:    address public gameSite;
7:    token public send;
8:    uint public itemPrice; uint public amount; uint public deadline; uint public limit;
9:    uint public moneyAccept;
10:
11:   enum GameRating { RatedAll, Rated12, Rated15, Rated18, ForTesting }
12:
13:   GameRating gameRatingChoice;
14:   GameRating constant defaultChoice = GameRating.Rated12;
15:
16:
17:   mapping(address => uint256) public balanceOf;
18:
19:   //생성자
20:   //부모가 계약을 생성할 때
21: function GameMoney(){
22:     parent = msg.sender;
23:     moneyAccept=0;
24: }
25: function setCondition(
26:   address gameSiteAddress, //게임사이트의 이더리움 주소
27:   uint item_price, //아이템 가격
28:   uint durationTime, //게임 머니 유효 기간
29:   uint limitMoney
30: ) onlyParent
31: {
32:   gameSite = gameSiteAddress;
33:   itemPrice = item_price * 1 ether;
34:   durationTime = deadline;
35: }
36:
```

```
function sendAmount(){
    eth.getBalance(parent) -= itemPrice
    eth.getBalace(child) += itemPrice
}

function returnAmount(){
    eth.getBalance(child) -= itemPrice
    eth.getBalace(parent) += itemPrice
}

function received(){
    deadline = now + deadline * 1 days;
    moneyAccept == 1;
}

modifier onlyParent(){
    if(msg.sender != parent)
    _;
}

modifier onlyChild(){
    if(msg.sender != child)
    _;
}

modifier afterDeadline() { if (now >= deadline) _; } //deadline 이후에는 게임 머니를 쓸 수 없음

function kill() {if(now >= deadline) selfdestruct(parent);} //deadline 이후에 블록 데이터가 모두 사라짐

function setGameRating(uint x) onlyParent{ //등급 설정하기
        if (x == 0) gameRatingChoice = GameRating.RatedAll;
        else if (x == 1) gameRatingChoice = GameRating.Rated12;
        else if (x == 2) gameRatingChoice = GameRating.Rated15;
        else if (x == 3) gameRatingChoice = GameRating.Rated18;
        else if (x == 4) gameRatingChoice = GameRating.ForTesting;
    }

function getGameRating() constant returns (uint) {
        return uint(gameRatingChoice);
    }

function getDeadline() constant returns (uint) {
        return uint(deadline);
}

function useGameMoney() payable onlyChild {
    send.transfer(gameSite, itemPrice);
}

function setEscrow() only gameSite //자녀가 사용하겠다고 표시하면 돈을 보내줌(에스크로)
        if(moneyAccept == 1) sendAmount();
        else returnAmount(); }}
```

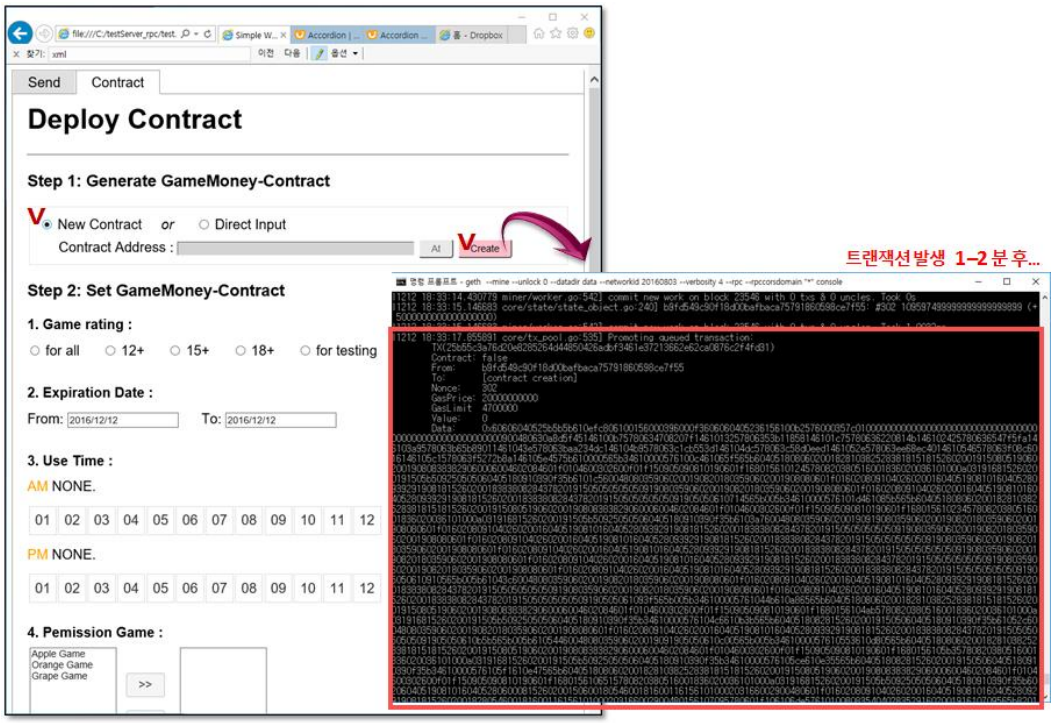
① 부모 지갑

부모 지갑은 새로운 계약 생성 또는 기존 계약을 불러와서 조건을 설정할 수 있다. 위에서 설계한 대로 게임 머니의 조건을 게임 등급, 유효 기간, 사용 시간 등을 부모가 설정하여 자녀의 주소로 게임 머니를 송금할 수 있다.



(그림 11) 부모 지갑 설계 화면

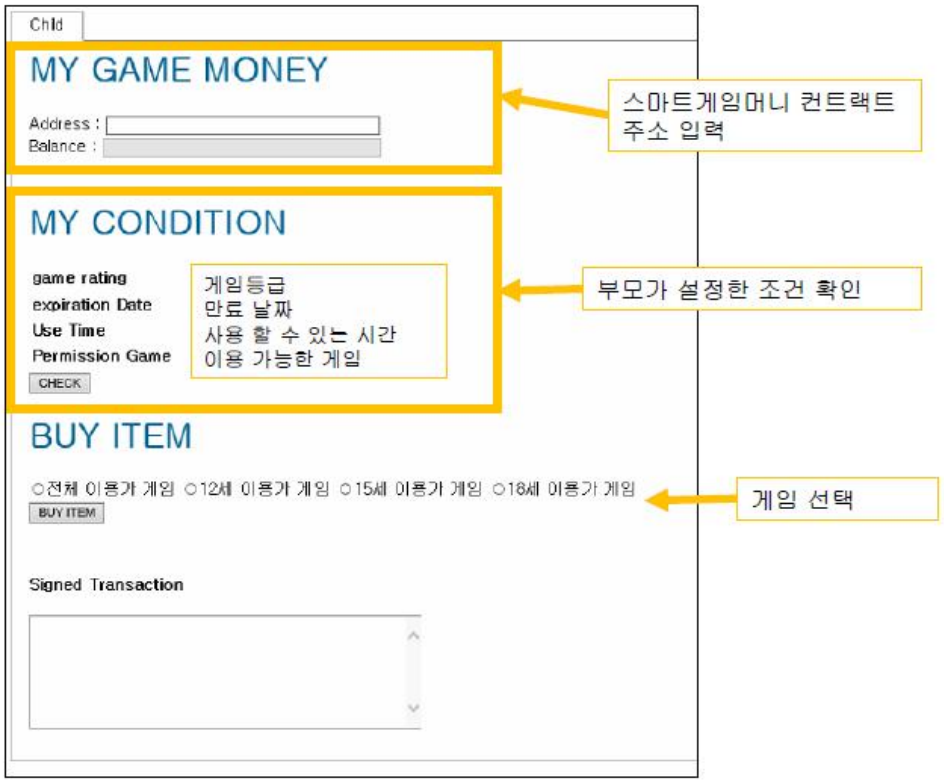
계약을 생성하는 방법은 새로운 계약 계정을 만들거나 기존의 스마트 계약의 조건만 수정하면 된다. 계약이 생성되고 조건이 변경될 때 마다 블록체인에 트랜잭션이 발생하기 때문에 언제 어떤 조건이 변경되었는지 확인이 가능하다.



(그림 12) 계약 생성 화면

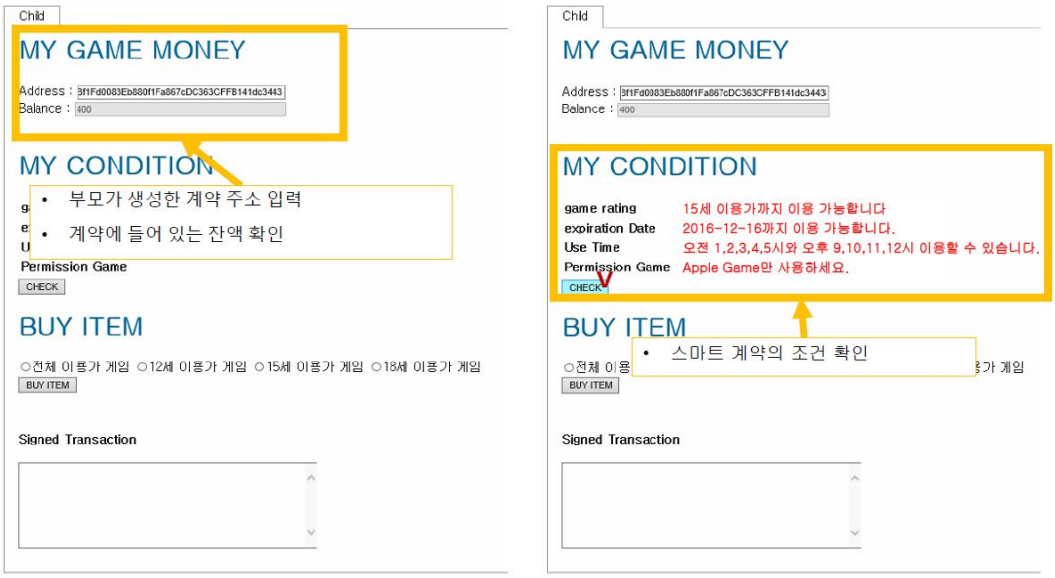
② 자녀 지갑 화면

자녀는 지갑을 통해 게임 머니를 받고, 부모가 설정한 조건을 확인을 할 수 있다.



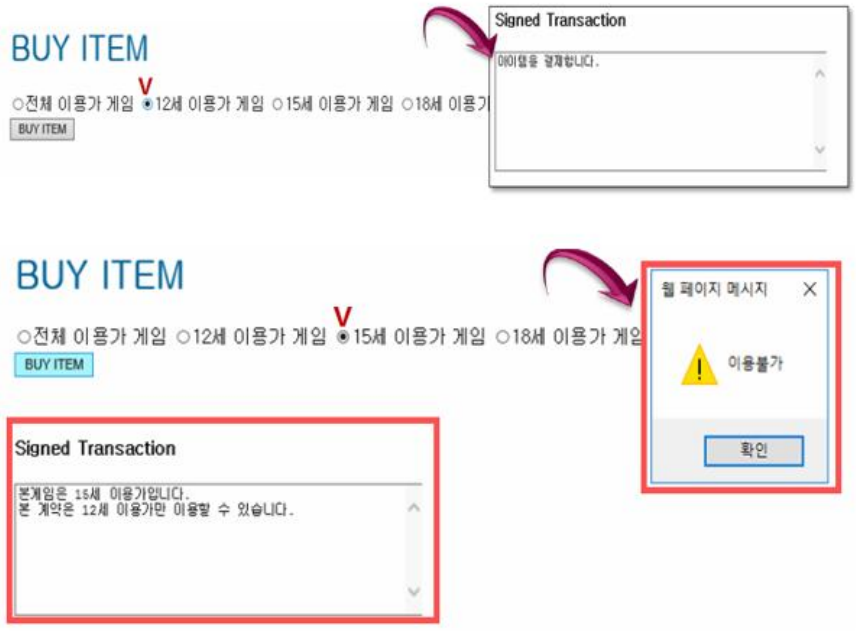
(그림 13) 자녀 지갑 설계 화면

게임 머니를 송금 받은 자녀는 자녀 지갑을 통해 계정의 주소를 입력하고 잔액을 확인할 수 있다. 조건 확인 창에서 부모가 송금 한 게임 등급, 만료 일자, 사용시간, 사용 가능 게임을 확인 할 수 있다. 위의 조건을 확인 한 후 게임을 선택하여 게임 머니를 사용할 수 있도록 구현해보았다.



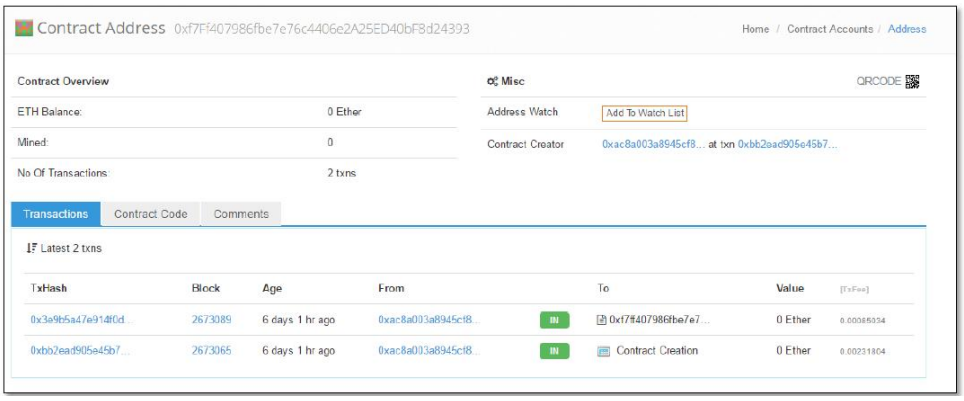
(그림 14) 자녀 지갑에서 계약 확인 화면

위 화면은 부모가 생성한 계약 계정을 입력하여 잔액과 게임 머니 조건을 확인하는 화면이다. 자녀는 부모가 설정한 조건을 수정할 수 없다.



(그림 15) 스마트 게임 머니 사용 화면

부모에게 받은 게임머니의 조건에 맞게 아이템을 구매하면 아이템이 결제가 되고 그렇지 않으면 콘솔창을 통해 계약에 대한 정보를 보여준다. 조건에 맞지 않아 결제가 되지 않을 때 트랜잭션을 생성하여 언제 게임 머니 사용을 시도했는지 timestamp를 통해 확인이 가능하다. 아래 그림은 실제 이더리움 탐색기에서 확인한 계약 계정의 트랜잭션 내역을 확인할 수 있다.



(그림 16) 계약 계정의 트랜잭션 확인

제 6 장 결론 및 향후 연구

블록체인은 중앙 서버가 존재하지 않고 자동으로 데이터가 이동이 되는 특징 때문에 다양한 분야에서 활용 가능성이 있다. 특히 블록체인의 스마트 계약은 실시간으로 계약을 생성하고 실행할 수 있기 때문에 금융, 유통, 보험 등 다양한 유형의 계약에 활용되고 있다. 하지만 계약에 들어가는 블록체인 사용자의 개인정보, 민감정보가 비식별화 조치없이 블록체인 네트워크에 올라가기 때문에 프라이버시의 위협이 될 가능성이 존재한다. 그리고 스마트 계약을 생성하기 위해서는 스마트 계약 코드를 작성하고 블록체인에 배치하기 위해서는 컴퓨터 코드에 대한 이해와 블록체인 네트워크에 대한 이해가 필요하기 때문에 스마트 계약을 통해 개인 간 거래를 하고 싶을 때 접근성이 떨어질 것으로 사료된다. 또한 블록체인이 국내에 알려진지 얼마 되지 않아 관련 법·제도에 대한 정보가 존재하지 않고 개인정보보호법에 명시되어 있는 개인정보 파기에 대한 내용이 위배가 되어 블록체인을 도입하기에는 미흡한 상황이다.

이에 본 논문에서는 블록체인을 활용한 스마트 계약은 블록체인의 핵심 기술로써, 스마트 계약을 활용해 신뢰도 및 대중성을 확보할 수 있지만 기존 중앙집중식 시스템과의 차이점을 고려한 블록체인 환경에서의 보안, 개인정보보호에 대한 부분이 미흡하다. 스마트 계약의 대중화를 위해서는 블록체인 내 신뢰도 향상이 필요하다 판단되어 신뢰할 수 있는 노드만 네트워크에 접근할 수 있도록 사용자 등록 시 type을 설정하여 type에 따라 접근제어를 할 수 있도록 설계하였다. 또 블록체인 기반 스마트 계약을 할 때 사용자 통합 인증을 통해 노드의 신뢰성을 확보하여 블록체인 노드의 정보보호를 위해 데이터 분류 및 비식별화 조치를 통해 안전한 블록체인 환경을 구축하고자 했다. 실생활에서 더 쉽게 스마트 계약을 활용할 수 있도록 스마트 계약 컨트롤러를 제안하며 일반 사용자들도 쉽게

스마트 계약을 실행하고 법률 언어와 함께 배치하여 법적 효력이 있는 계약을 개발하고자 했다.

향후 연구로는 본 연구에서 제시한 신뢰할 수 있는 블록체인 설계 방안처럼 사용자 등록과 통합 인증에 대한 절차를 세분화하여 노드의 신뢰성을 높이고 데이터 분류를 더 세분화하여 블록에 최소한의 데이터만 저장할 수 있는 방안에 대해 연구하고자 한다. 또한 계약 목적, 유형을 선택하면 거기에 맞는 템플릿만 채우면 계약이 진행되는 스마트 계약 컨트롤러를 구축하고자 한다. 이를 위해 각 계약서에 대한 관련연구와 스마트 계약에 대한 연구를 지속적으로 수행할 예정이다. 아울러 본 논문은 이더리움 플랫폼을 기반으로 제안되었다. 이를 보완하기 위해 Hyperledger의 Fabric 등 다양한 플랫폼에서 설계 방안을 검증하여 새로운 블록체인 네트워크를 구축하여 신뢰할 수 있는 블록체인을 설계 하기 위한 연구를 계속 할 예정이다.

참 고 문 헌

[1] 한국은행 금융결제국. “분산원장 기술과 디지털통화의 현황 및 시사점”. 한국은행. 2016

[2] 한국정보화진흥원, “Beyond 비트코인, 블록체인 기술의 무한확장”, IT&Future Strategy 제 9호, 2016

[3] 보안연구부 . “금융권 특화 블록체인 플랫폼 Corda의 주요 특징 소개”. 금융. 2017

[4] 이혁준, 이수미. “비트코인의 신뢰구조와 이중지불의 위협”. 정보보호학회지, 26(2), 25-30. 2016

[5] 신다혜, 이종협. “핀테크를 위한 스마트 컨트랙트 보안”. 정보처리학회지, v.22. no.5, pp. 54-62. 2015

[6] Juels, Ari, Ahmed Kosba, and Elaine Shi. "The ring of gyges: Using smart contracts for crime." aries 40 (2015): 54.

[7] CLACK, Christopher D.; BAKSHI, Vikram A.; BRAINE, Lee. Smart Contract Templates: foundations, design landscape and research directions. arXiv preprint arXiv:1608.00771, 2016.

[8] 유헌우, “블록체인 방식의 전자투표 시스템 구현 및 성능 개선 방안 연구”, 아주대학교 정보통신대학원, 2016

[9] 문정환. “블록체인 기반 쿠폰 서비스 연구”, 동국대학교 국제정보대학원, 2017

[10] 오서영, 이창훈, “부동산 시장의 신뢰성 향상을 위한 블록체인 응용 기술”, 한국전자거래학회지, Vol.22 No.1, 2017

[11] Fido Alliance. https://fido.kica.co.kr/. 2017.3

[12] 개인정보보호 종합포털. “개인정보 비식별 조치 가이드라인”.

https://www.privacy.go.kr
[13] 한국CPO포럼. “CPPG가이드북”. 2012.
[14] “청소년 ‘인터넷 중독’ 심각...고위험군 일본의 4.4배, 위클리 오늘, 2017년 3월 19일, http://www.weeklyoday.com/news/articleView.html?idxno=551 58.
[15] “친구들에 ‘무시’ 당하기 싫어 게임 과외까지 받는 청소년들”, Insight, 2017년 4월 24일, http://www.insight.co.kr/newsRead.php?ArtNo=102257
[16] “청소년 ‘19금 앱’ 사용 규제 허술, 국제신문, 2016년 10월 17일, 20면
[17] “게임에 중독돼 200만원 결제”, 강원일보, 2017년 2월 2일, 5면
[18] 미래창조과학부, 정보화진흥원. “2016년 인터넷 과의존 실태조사 결과”, 2017
[19] 국가법령정보센터, www.law.go.kr, 2017
[20] 홍승필, “개인정보보호 개론”, 한티미디어, 2009

ABSTRACT

Design and Application of Reliable Blockchain in Digital Business Environment

Park Sumin
Dept. of Computer Science
The Graduate School
Sungshin Women's University

ICT services combining big data, IoT, and artificial intelligence are attracting attention. In order to distribute and store a large amount of data, the blockchain has attracted attention as a component technology of the fourth industrial revolution. The blockchain is distributed as a distributed system without a 'trusted third party', making it a very secure technology because it is hard for hackers to hack the data. In particular, a smart contract based on a blockchain is a technology that creates a trusted contract environment in a P2P environment. By establishing conditions and creating a contract through a contract code, it is possible to execute sophisticated contracts between contract parties, It is a technology that companies are paying attention to. At present, many companies in Korea and abroad have developed their own block chain platform to utilize blockchain to provide new services and utilize them in various fields.

However, since block-chain technology is code-based, it is difficult for ordinary

people who do not understand the blockchain to utilize smart contracts. And since block chains have the same information, they guarantee integrity but do not provide confidentiality for data protection. Therefore, when personal information or sensitive information is not protected and uploaded to the blockchain, the information is high probability of exposure. In addition, since a blockchain can not be canceled once a block is created, it is considered necessary to perform a non-idenification measure before putting data on the blockchain.

In this paper, I suggest a reliable blockchain design method, and build a smart contract environment based on blockchain in a secure environment. The design includes functions of access control through data classification, and smart contract controller that easily parses contract codes is developed based on the problems that can occur in the blockchain environment. In order to verify the applicability of the proposed design method in real environment, I suggest the future research method of case study and design method of smart game money based on blockchain.

별 첨

Wallet

```
<html lang="en">
<head>
 <meta charset="UTF-8">

 <!-- JS -->
 <script type="text/javascript" src="https://code.jquery.com/jquery-1.12.4.js"></script>
 <script type="text/javascript" src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>
    <script type="text/javascript" src="./bower_components/web3/dist/web3.js"></script>
 <script type="text/javascript" src="./lib_js/wallet-design.js"></script>
 <script type="text/javascript" src="./lib_js/wallet-contract-permissionGame.js"></script>

 <!-- CSS -->
 <link rel="stylesheet" href="./lib_css/tap.css" />
 <link rel="stylesheet" href="./lib_css/body.css" />
 <link rel="stylesheet" href="https://code.jquery.com/ui/1.12.1/themes/base/jquery-ui.css">

 <title>Simple Wallet</title>
</head>

<body>
 <div id="container">
  <ul class="tab">
   <li class="active"><a href="#tab1">Send</a></li>
   <li><a href="#tab2">Contract</a></li>
  </ul>

  <div class="tab_container">
   <div class="tab_content" id="tab1" style="display: block;">
    <!--Content-->
    <h1>Send Ether</h1>
    <hr>
    <P>
     <h3>Step 1: Generate Information</h3>
     From Address : <input type="text" id="fromAddress" size="46" /><br />
     Balance : <input type="text" id="balance" size="47" disabled="disabled"/> ETH
    </p>
    <P>
     <h3>Step 2: Send Transaction</h3>
     To Address : <input type="text" id="toAddress" size="50" /><br />
     Amount to Send : <input type="text" id="amount" size="36" /> ETH
    </p>
    <button id="sendAmount" onclick='javascript:sendAmount();'>Send</button>
    <br /><br />
   </div>

   <div class="tab_content" id="tab2" style="display: none;">
     <!--Content-->
    <h1>Deploy Contract</h1>
```

```
    <hr>
    <P> <!-- Contract 생성 -->
     <h3>Step 1: Generate GameMoney-Contract</h3>

      <fieldset>
      <br style="line-height:10px" /> 
      <input type="radio" name="radio-1" id="radio-1" value="0"
onclick="javascript:checkContract(0);" />
      New Contract    <i> or </i>    
      <input type="radio" name="radio-1" id="radio-2" value="1"
onclick="javascript:checkContract(1);" />
      Direct Input <br />

      <div style="line-height:40px">
             
       Contract Address : <input type="text" id="contractAddress" size="50" />
       <button id='atContract' onclick='javascript:atContract();'
style="line-height:12px">At</button>
       <button id='createContract' onclick='javascript:createContract();'
style="line-height:12px">Create</button>
      </div>

      </fieldset>
    </p>

    <p> <!-- Contract 조건설정 -->
     <h3>Step 2: Set GameMoney-Contract</h3>

     <!-- 게임등급 -->
     <b>1. Game rating : </b><br />
     <div style="line-height:50px">
      <input type="radio" name="gameRating" value="0"
onclick="javascript:checkGameRating(0);"/> for all     
      <input type="radio" name="gameRating" value="1"
onclick="javascript:checkGameRating(1);"/> 12+     
      <input type="radio" name="gameRating" value="2"
onclick="javascript:checkGameRating(2);"/> 15+     
      <input type="radio" name="gameRating" value="3"
onclick="javascript:checkGameRating(3);"/> 18+     
      <input type="radio" name="gameRating" value="4"
onclick="javascript:checkGameRating(4);"/> for testing
      <br />
     </div>
     <br />

     <!-- 유효기간 -->
     <b>2. Expiration Date : </b><br />
     <div style="line-height:50px">
      <script>
      $(function() {
       $("#datepickerFrom").datepicker({
        dateFormat:"yy/mm/dd"
       }).datepicker("setDate",new Date());
       $("#datepickerTo").datepicker({
        dateFormat:"yy/mm/dd"
```

```
        }).datepicker("setDate",new Date());
      });
      </script>
      From: <input type="text" id="datepickerFrom" title="오늘날짜 이후로 선택"/>
    
      To: <input type="text" id="datepickerTo" title="오늘날짜 이후로 선택"/>
      <br />
    </div>
    <br />

    <!-- 사용시간 -->
    <b>3. Use Time : </b><br />
    <div style="line-height:50px">
     <script>
       var selectTimeAM = null;
       var selectTimePM = null;
       $( function() {
      $( "#selectableAM" ).selectable({
        stop: function() {
       var result = $( "#select-resultAM" ).empty();

       $( ".ui-selected", this ).each(function() {
         var index = $( "#selectableAM td" ).index( this );
         if (index != -1)
        result.append( " " + ( index + 1 ) + "시" );
         selectTimeAM = result.text();
       });
        }
      });

      $( "#selectablePM" ).selectable({
        stop: function() {
       var result = $( "#select-resultPM" ).empty();
       $( ".ui-selected", this ).each(function() {
         var index = $( "#selectablePM td" ).index( this );
         if (index != -1)
        result.append( " " + ( index + 1 ) + "시" );
         selectTimePM = result.text();
       });
        }
      });
       } );
    </script>
    <font color="orange">AM</font> <span id="select-resultAM"> NONE</span>.
    <table id="selectableAM">
     <tr>
      <td class="ui-widget-content">01</td>
      <td class="ui-widget-content">02</td>
      <td class="ui-widget-content">03</td>
      <td class="ui-widget-content">04</td>
      <td class="ui-widget-content">05</td>
      <td class="ui-widget-content">06</td>
      <td class="ui-widget-content">07</td>
      <td class="ui-widget-content">08</td>
      <td class="ui-widget-content">09</td>
```

```
          <td class="ui-widget-content">10</td>
          <td class="ui-widget-content">11</td>
          <td class="ui-widget-content">12</td>
         </tr>
        </table>

        <font color="orange">PM</font> <span id="select-resultPM"> NONE</span>.
        <table id="selectablePM">
         <tr>
          <td class="ui-widget-content">01</td>
          <td class="ui-widget-content">02</td>
          <td class="ui-widget-content">03</td>
          <td class="ui-widget-content">04</td>
          <td class="ui-widget-content">05</td>
          <td class="ui-widget-content">06</td>
          <td class="ui-widget-content">07</td>
          <td class="ui-widget-content">08</td>
          <td class="ui-widget-content">09</td>
          <td class="ui-widget-content">10</td>
          <td class="ui-widget-content">11</td>
          <td class="ui-widget-content">12</td>
         </tr>
        </table>
       </div>
       <br />

       <!-- 허용게임 -->
       <div style="line-height:35px">
        <label for="speed"><b>4. Pemission Game : </b></label>
        <table>
         <tr>
          <td>
          <div style="width:130px;float:left">
           <select size="10" multiple="multiple" name="availableItems"
id="availableItems" style="width:120px;" title="선택할 수 있는 게임목록">
            <option value="0">Apple Game</option>
            <option value="1">Orange Game</option>
            <option value="2">Grape Game</option>
           </select>
          </div>
          </td>

          <td>
          <div style="width:70px;">

            <button class="ui-button ui-widget ui-corner-all btn" onclick="addItems();"
>&gt;&gt;</button><br /><br />
            <button class="ui-button ui-widget ui-corner-all btn" onclick="removeItems();"
>&lt;&lt;</button>
          </div>
          </td>

          <td>
          <div style="width:130px;">
           <select size="10" multiple="multiple" name="selectedItems" id="selectedItems"
```

```
title="선택한 게임목록" style="width:120px;">
          </select>
         </div>
         </td>
        </tr>
       </table>
      </div>

      <!-- 머니송금 -->
      <div style="line-height:30px">
      <br />
       <b>5. Amount to Send</b> : <input type="text" id="gamemoneyAmount"
size="36" title="게임머니로 전송금액 입력" /> ETH
      </div>

      <!-- 확인 창 -->
      <div id="dialog" title="Gamemoney Set Up" >
      </div>
      <br />

      <script>
      $( function() {
       $( "#accordion" ).accordion({
         heightStyle: "content",
         active: 3
       });
      } );
      </script>

      <script>
        $( function() {
       function log( message ) {
         //$( "<div/>" ).text( message ).prependTo( "#log" );
         //$( "#log" ).attr( "scrollTop", 0 );
       }
       $.ajax({ /* 아직 만들지 않았어요.. */
         url: "london.xml",
         dataType: "xml",
         success: function( xmlResponse ) {
        var data = $( "geoname", xmlResponse ).map(function() {
          return {
         value: $( "name", this ).text() + ", " +
           ( $.trim( $( "countryName", this ).text() ) || "(unknown country)" ),
         id: $( "geonameId", this ).text()
          };
        }).get();

        $( "#birds" ).autocomplete({
          source: data,
          minLength: 0,
          select: function( event, ui ) {
         log( ui.item ?
           "Selected: " + ui.item.value + ", geonameId: " + ui.item.id :
           "Nothing selected, input was " + this.value );
          }
```

```
        });
          }
       });
        } );
      </script>

      <div id="accordion">

        <h3>[Option] 6. 송금허용 </h3>
        <div style="line-height:35px;font-size:15px">
       <input type="text" id="sendPartAddress" size="36" title="누구에게"
/> 에게 <br />
       <input type="radio" name="sendPart" value="0" /> 금지
     
       <input type="radio" name="sendPart" value="1" /> 전체의
       <input type="text" id="sendPartAmount" size="5" title="입력된 퍼센트까지만
송금허용" /> % 부분허용      
       <input type="radio" name="sendPart" value="2" /> 전체허용
       <br />
        </div>

        <h3>[Option] 7. 스마트게임 외 사용처 </h3>
        <div class="ui-widget" style="line-height:35px;font-size:15px">
         사용처 찾기 :  <input id="birds" title="사용할 수 있는 곳 지정"/>
<button id='searchUse' class="ui-button ui-widget ui-corner-all btn" style="height: 28px;
" >*</button> <br />
         사용할 수 있는 곳(제휴처 등) :
         <div style="height: 150px; width: 350px; overflow: auto;line-height:20px;"
class="ui-widget-content">
        카카오프렌즈 온라인스토어 : 캐릭터구매 <br />
        라인 캐릭터샵 : 캐릭터구매
         </div>
        </div>

        <h3>[Option] 8. 사용한도 </h3>
        <div style="line-height:35px;font-size:15px">
       1회사용 한도 :
       <input type="text" id="onetimeLimit" size="15" title="1회사용 한도 입력"
/> ETH <br />
       1일사용 한도 :
       <input type="text" id="dailyLimit" size="15" title="1일사용 한도 입력"
/> ETH <br />
        </div>

      </div>
      <br />


      <button id='initGameRating' onclick='javascript:if(confirm("게임머니도
전송하시겠습니까?")){setGameMoneyValues();}'>Save and Send</button>
      <button id='setGameRating' onclick='javascript:setGameRating();'>Save</button>
      <button id='getGameRating' onclick='javascript:getGameMoneyValues();'>Call
values</button>
    </p>
   </div>
```

```
    <!-- 실시간 이벤트 확인 -->
    <div style="padding: 20px;" >
     <P>
      <h3>Signed Transaction</h3>
      <textarea id="log" cols="52" rows="8" ></textarea>
     </p>
    </div>

   </div>
  </div>
</body>

<script type="text/javascript">
   var Web3 = require('web3');
   var web3 = new Web3(new Web3.providers.HttpProvider("http://localhost:8545"));
   //var web3 = new Web3(new
Web3.providers.IpcProvider("\\\\.\\pipe\\C:/Users/islab/AppData/Roaming/Ethereum/geth.ipc",
require('net')));
   var logColor = 0;

   // Send //////////////////////////////
   var fromAddress = web3.eth.accounts[0];
   var balance = web3.fromWei(web3.eth.getBalance(fromAddress), "ether");

   $('#fromAddress').val(fromAddress);
   $('#balance').val(balance);

   var toAddress = web3.eth.accounts[1];
   $('#toAddress').val(toAddress);

   function sendAmount() {
    logColor = 0;
    var amount = web3.toWei($('#amount').val(), "ether");
    web3.eth.sendTransaction({
     from:web3.eth.accounts[0],
     to:toAddress,
     value:amount
    });
   }

   // Contract ////////////////////////////
   var gameRating = null;
   var contractAddress = null;

   var gamemoneyContract =
web3.eth.contract([{"constant":true,"inputs":[],"name":"getExpirationDateFrom","outputs":[{"na
me":"","type":"string"}],"payable":false,"type":"function"},{"constant":false,"inputs":[{"name":"_a
m","type":"string"},{"name":"_pm","type":"string"}],"name":"setSelectTime","outputs":[],"payable"
:true,"type":"function"},{"constant":true,"inputs":[],"name":"getSelectTimeAM","outputs":[{"name
":"","type":"string"}],"payable":false,"type":"function"},{"constant":false,"inputs":[{"name":"_gr","
type":"uint256"},{"name":"_from","type":"string"},{"name":"_to","type":"string"},{"name":"_am"
,"type":"string"},{"name":"_pm","type":"string"},{"name":"_list","type":"string"}],"name":"setGa
meMoneyValues","outputs":[],"payable":true,"type":"function"},{"constant":false,"inputs":[{"name
":"_from","type":"string"},{"name":"_to","type":"string"}],"name":"setExpirationDate","outputs":[]
```

,"payable":true,"type":"function"},{"constant":true,"inputs":[],"name":"getPermissionGameList","outputs":[{"name":"","type":"string"}],"payable":false,"type":"function"},{"constant":true,"inputs":[],"name":"getGameRating","outputs":[{"name":"","type":"uint256"}],"payable":false,"type":"function"},{"constant":false,"inputs":[{"name":"_list","type":"string"}],"name":"setPermissionGameList","outputs":[],"payable":true,"type":"function"},{"constant":false,"inputs":[{"name":"x","type":"uint256"}],"name":"setGameRating","outputs":[],"payable":true,"type":"function"},{"constant":true,"inputs":[],"name":"getSelectTimePM","outputs":[{"name":"","type":"string"}],"payable":false,"type":"function"},{"constant":true,"inputs":[],"name":"getDefaultGameRating","outputs":[{"name":"","type":"uint256"}],"payable":false,"type":"function"},{"constant":true,"inputs":[],"name":"getExpirationDateTo","outputs":[{"name":"","type":"string"}],"payable":false,"type":"function"},{"inputs":[],"payable":true,"type":"constructor"}]);

  // New Contract
  function createContract() {
   logColor = 1;

   gamemoney = gamemoneyContract.new({
    from: web3.eth.accounts[0],
    data:
'0x60606040525b5b5b610efc806100156000396000f360606040523615610 0b2576000357c01
0000000000000000000000000000000000000000000000000000000900480630a8d5f45146
100b75780634708207f1461013257806353b11858146101c75780636220814b146102425780
636547f5fa146103a9578063b65b89011461043e578063baa234dc146104b9578063c1cb553d1
46104dc578063c58d0eed1461052e578063ee68ec4014610546578063f08c6016146105c15780
63f5272b8a146105e4575b610000565b34610000576100c461065f565b604051808060200182
8103825283818151815260200191508051906020019080838382906000600460208460 1f010
4600302600f01f150905090810190601f1680156101245780820380516001836020036101000
a031916815260200191505b509250505060405180910390f35b6101c5600480803590602001
9082018035906020019080806 01f0160208091040260200160405190810160405280939291908
1815260200183838082843782019150505050505091908035906020019082018035906020019
0808060 1f0160208091040260200160405190810160405280939291908181526020018383808
2843782019150505050505091905050 61714565b005b34610000576101d461085b565
b604051808060200182810382528381815181526020019150805190602001908083838290600
0060046020846 01f0104600302600f01f150905090810190601f1680156102345780820380516
0018360200361010 00a031916815260200191505b509250505060405180910390f35b6103
a760048080359060200190919080359060200190820180359060200190808060 1f0160208091
0402602001604051908101604052809392919081815260200183838082843782019150505
0505050919080359060200190820180359060200190808060 1f0160208091040260200160405
1908101604052809392919081815260200183838082843782019150505050505091908035
9060200190820180359060200190808060 1f0160208091040260200160405190810160405280
9392919081815260200183838082843782019150505050505091908035906020019082018
0359060200190808060 1f0160208091040260200160405190810160405280939291908181526
0200183838082843782019150505050505091908035906020019082018035906020019080
80601f0160208091040260200160405190810160405280939291908181526020018383808284
37820191505050505050919050506 10910565b005b61043c6004808035906020019082018
0359060200190808060 1f0160208091040260200160405190810160405280939291908181526
0200183838082843782019150505050505091908035906020019082018035906020019080
8 0601f0160208091040260200160405190810160405280939291908181526020018383808284
37820191505050505050919050506 1093f565b005b346100005761044b610a86565b6040518
0806020018281038252838181518152602001915080519060200190808383829060006004 6
02084601f0104600302600f01f150905090810190601f1680156104ab5780820380516001836
020036101000a031916815260200191505b509250505060405180910390f35b346100005761
04c6610b3b565b604051808281526020019150506040518091039 0f35b61052c600480803590
6020019082018035906020019080806 01f0160208091040260200160405190810160405280939
2919081815260200183838082843782019150505050505091905050610b5b565b005b6105

4460048080359060200190919050506 10c00565b005b3461000057610553610d80565b60405
1808060200182810382528381815181526020019150805190602001908083838290600060 0
4602084601f0104600302600f01f150905090810190601f1680156105b357808203805160018
36020036101000a0319168152602001915050b509250505050604051809103 90f35b3461000057
6105ce610e35565b60405180828152602001915050604051809103 90f35b34610000576105f1
610e47565b60405180806020018281038252838181518152602001915080519060200190808
38382906000600460208460 1f0104600302600f01f150905090810190601f168015610651578
082038051600183602003 6101000a031916815260200191505b509250505050604051809103 9
0f35b602060405190810160405280600081526020015060018054600181600116156101000 2
03166002900480601f0160208091040260200160405190810160405280929190818152602 00
1828054600181600116156101000203166002900480156107095780601f106106de57610100
8083540402835291602001916107 09565b820191906000526020600020905b815481529060
0101906020018083116106ec57829003601f168201915b505050505090505b90565b8160039
0805190602001908280546001816001161561010002031660029004906000526020600020 9
0601f016020900481019282601f10610760578051 60ff19168380011785556107 8e565b82800
160010185555821561078e579182015b828111156107 8d578251825591602001919060010190
610772565b5b5090506107b391905b80821115610 7af5760008160009055506001016107975
65b5090565b50508060049080519060200190828054600181600116156101000203166002 9
0049060005260206000209060 1f016020900481019282601f10610801578051 60ff191683800
117855561082f565b8280016001018555582156108 2f579182015b828111156108 2e578251825
5916020019190600101906108 13565b5b50905061085491905b808211156108505760008160
009055506001016108 38565b5090565b50505b5050565b6020604051908101604052806000 8
15260200150600380546001816001161561010002031660029004806 01f0160208091040260
2001604051908101604052809291908181526020018280546001816001161561010002031 6
60029004801561090557806 01f106108da5761010080835404028352916020019161090556 5
b820191906000526020600020905b81548152906001019060200180831161 08e8578290036 0
1f168201915b505050505090505b90565b610919866 10c00565b61092385856109 3f565b6109
2d838361071456 5b6109368161 0b5b565b5b505050505050565b816001908051906020019 08
280546001816001161561010002031660029004906000526020600020906 01f016020900481
019282601f1061098b578051 60ff19168380011785556109b9565b828001600101855558215 61
09b9579182015b82811115610 9b8578251825591602001919060010190610 99d565b5b50905
06109de91905b80821115610 9da5760008160009055506001016109c2565b5090565b505080
6002908051906020019082805460018160011615610100020316600290049060005260206 0
00209060 1f016020900481019282601f10610a2c578051 60ff1916838001178555610a5a565b
8280016001018555821561 0a5a579182015b82811115610a59578251825591602001919060
010190610a3e565b5b509050610a7f91905b80821115610a7b5760008160009055506001016
10a63565b5090565b50505b5050565b602060405190810160405280600081526020015060 05
8054600181600116156101000203166002900480601f01602080910402602001604051908 10
1604052809291908181526020018280546001816001161561010002031660029004801561 0
b305780601f10610b0557610100808354040283529160200191610b30565b820191906000 52
6020600020905b8154815290600101906020018083116 10b1357829003601f168201915b505
0505050 90505b90565b6000600060149054906101000a900460ff166004811161 0000579050
5b90565b8060059080519060200190828054600181600116156101000203166002900490 60
0052602060002090601f016020900481019282601f10610ba757805160ff1916838001178555
610bd5565b82800160010185555821561 0bd5579182015b82811115610bd457825182559160 2
00191906001019 0610bb9565b5b509050610bfa91905b80821115610bf65760008160009055 5
0600101610bde565b5090565b50505b50565b600081141561 0c4c5760006000601461 01000a
81548160ff02191690837f010000000000000000000000000000000000000000000000000000
0000000000908102040217905550610d7c565b600181141561 0c985760016000601461 01000
a81548160ff02191690837f01000000000000000000000000000000000000000000000000000
00000000000908102040217905550610d7b565b600281141561 0ce4576002600060146101 00
0a81548160ff02191690837f0100000000000000000000000000000000000000000000000000
000000000000908102040217905550610d7a565b600381141561 0d3057600360006014 6101
000a81548160ff02191690837f010000000000000000000000000000000000000000000000000
000000000000090810204021790555061 0d79565b600481141561 0d78576004600060 1461

```
01000a81548160ff02191690837f010000000000000000000000000000000000000000000000000000
0000000000000000908102040217905550 5b5b5b5b5b5b50565b602060405190810160405280
06000815260200150600480546001816001161561010002031660029004806 01f0160208091
040260200160405190810160405280929190818152602001828054600181600116156101000
020316600290048015610e2a5780601f10610dff576101008083540402835291602001916 10e
2a565b820191906000526020600020905b81548152906001019060200180831161 0e0d57829
003601f168201915b505050505090505b90565b600060016004811161000579050 5b90565b
602060405190810160405280600081526020015060028054600181600116156101000203 16
6002900480601f016020809104026020016040519081016040528092919081815260200182 8
054600181600116156101000203166002900480 15610ef15780601f10610ec65761010080835
404028352916020019161 0ef1565b820191906000526020600020905b815481529060010190
602001808311610ed457829003601f168201915b505050505090505b9056',
    gas: '4700000' },
   function (e, contract){
    console.log(e, contract);
    if (typeof contract.address !== 'undefined') {
      console.log('Contract mined! address: ' + contract.address + ' transactionHash: '
+ contract.transactionHash);

      alert ('Contract mined! \n' + contract.address);
      $('#log').append('<font color="hotpink">Contract mined! address: ' +
contract.address + ' transactionHash: ' + contract.transactionHash + '</font>\n\n');

      $('#contractAddress').val(contract.address);

      contractAddress = contract.address;
    }
    });
  }

  // Direct Input
  function atContract() {
   contractAddress = $('#contractAddress').val();
   gamemoney = gamemoneyContract.at(contractAddress);
   if (gamemoney != null)
    alert("게임머니 선택되었습니다.");
  }

  function checkContract(chk) {
   if (chk == 0) {
    $("#contractAddress").attr("readOnly", true);
    $("#contractAddress").attr("style", "border-width:1px; border-collapse:collapse;
background-color:lightgray; border-bottom-style:inset;");
    $("#atContract").attr("disabled", true);
    $("#atContract").attr("style", "line-height:12px;background-color:#EAEAEA");
    $("#createContract").attr("disabled", false);
    $("#createContract").attr("style", "line-height:12px");

   } else {
    $("#contractAddress").attr("readOnly", false);
    $("#contractAddress").attr("style", "");
    $("#atContract").attr("disabled", false);
    $("#atContract").attr("style", "line-height:12px;");
    $("#createContract").attr("disabled", true);
    $("#createContract").attr("style", "line-height:12px;background-color:#EAEAEA");
```

```
 }

}

// 게임등급 선택 ////////////////////////////
function checkGameRating(rated) {
 gameRating = rated;
}

function msgGameRating(gr) {
 var msg = "<b>게임등급</b>은 " + "<font color='deepskyblue'>";

 switch (Number(gr)) {
  case 0:
   msg += "전체 이용가";
   break;
  case 1:
   msg += "12세 이용가";
   break;
  case 2:
   msg += "15세 이용가";
   break;
  case 3:
   msg += "18세 이용가";
   break;
  default:
   msg += "테스트용";
 }

 msg += "</font> 사용.<br /><br />";

 return msg;
}

function setGameRating() {
 logColor = 2;
 gamemoney.setGameRating(gameRating, {from: web3.eth.accounts[0]});
}

function getGameRating() {
 var gameRatingMsg = null;

 if (gameRating == null) {
  gameRatingMsg = msgGameRating(gamemoney.getDefaultGameRating());
 } else {
  gameRatingMsg = msgGameRating(gamemoney.getGameRating());
 }
 return gameRatingMsg;
}

// 유효기간 선택 ////////////////////////////
var dateFrom = null;
var dateTo = null;
function checkExpirationDate(tr) {
```

```
    dateFrom = $("#datepickerFrom").val();
    dateTo = $("#datepickerTo").val();

    if (!tr) {
     dateFrom = "2016/12/01";
     dateTo = "2016/12/31";
    } else {
     dateFrom = gamemoney.getExpirationDateFrom();
     dateTo = gamemoney.getExpirationDateTo();
    }

    var expirationDate = "<b>유효기간</b>은 <br />" +
          "<font color='deepskyblue'>"+ dateFrom + "</font> 부터 " +
          "<font color='deepskyblue'>"+ dateTo + "</font> 까지 사용. <br /><br
/>";

    return expirationDate;
   }

   // 사용시간 선택 ////////////////////////////
   //var useTimeAM = [];
   /*
   function checkSplit(str) {
    var arrChange = [];
    var arrStr = str.split(" ");
    for (var i = 0; i< arrStr.length-1; i++) {
     arrChange.push([parseInt(arrStr[i])]);
    }
    return arrChange;
   }*/

   function checkUseTime(tr) {
    var utAM = "";
    var utPM = "";

    if(!tr) {
     utAM = useTimeAM();
     utPM = useTimePM();
    } else {
     utAM = gamemoney.getSelectTimeAM();
     utPM = gamemoney.getSelectTimePM();
    }

    var useTime = "<b>사용시간</b>은 <br />" +
         "오전 " + "<font color='deepskyblue'>"+ utAM + "</font> 선택<br />" +
         "오후 " + "<font color='deepskyblue'>"+ utPM + "</font> 선택<br /><br
/>";

    return useTime;
   }

   function useTimeAM() {
    var selectTime = "";

    if (selectTimeAM == null) {
```

```
        selectTime = "NONE";
       } else {
        var arrStr = selectTimeAM.split(" ");
        for (var i = 0; i< arrStr.length-1; i++) {
         if (selectTimeAM != "")
          selectTime = selectTimeAM + " ";
        }
       }

       return selectTime;
      }

      function useTimePM() {
       var selectTime = "";

       if (selectTimePM == null) {
        selectTime = "NONE";
       } else {
        var arrStr2 = selectTimePM.split(" ");
        for (var i = 0; i< arrStr2.length-1; i++) {
         if (selectTimePM != "")
          selectTime = selectTimePM + " ";
        }
       }

       return selectTime;
      }

      // 조건설정 & 게임머니 송금 ////////////////////////
      function setGameMoneyValues() {
       logColor = 2;

       // 유효기간
       dateFrom = $("#datepickerFrom").val();
       dateTo = $("#datepickerTo").val();

       // 사용시간
       var utAM = useTimeAM();
       var utPM = useTimePM();

       // 허용게임 목록
       frmSubmit(false);

       // 송금금액
       var gamemoneyAmount = web3.toWei($('#gamemoneyAmount').val(), "ether");

       gamemoney.setGameMoneyValues(gameRating, dateFrom, dateTo, utAM, utPM,
    gameList, web3.eth.accounts[0], {
        from: web3.eth.accounts[0],
        value: gamemoneyAmount,
        gas: 1000000,
       });
      }

      // 확인 창 ////////////////////////////////
```

```
    function getGameMoneyValues() {
     logColor = 2;
     if (gameRating == null) {
      var msg = msgGameRating(gamemoney.getDefaultGameRating());
      var msg2 = checkUseTime(false);
      var msg3 = checkExpirationDate(false);
      var msg4 = frmSubmit(false);
      $( "#dialog" ).html(msg + msg2 + msg3 + msg4);
      $( "#dialog" ).dialog( "option", "width", 350 );
      $( "#dialog" ).dialog( "open" );

     } else {
      var msg = msgGameRating(gamemoney.getGameRating());
      var msg2 = checkUseTime(true);
      var msg3 = checkExpirationDate(true);
      var msg4 = frmSubmit(true);
      $( "#dialog" ).html(msg + msg2 + msg3 + msg4);
      $( "#dialog" ).dialog( "option", "width", 350 );
      $( "#dialog" ).dialog( "open" );
     }
    }

    $( "#dialog" ).dialog({
     autoOpen: false,
     modal: true,
     buttons: {
      Ok: function() {
       $( this ).dialog( "close" );
      }
     }
    });
   </script>
   <script type="text/javascript" src="./lib_js/wallet-logEvent.js"></script>
  </html>
```