분산 모바일 어플리케이션 플랫폼의 스마트-계약 가치-전송 프로토콜

Patirick Dai[1], Neil Mahi[1], Jordan Earls[1], Alex Norta[2]

[1] Qtum Foundation Singapore

foundation@qtum.org

[2] Large-Scale Systems Group, Tallinn University of Technology,

Akadeemia tee 15A, 12816 Tallinn, Estonia

Alex.norta.phd@ieee.org

초록

트랜잭션에 대한 증명 확인 검증, 블록 체인을 사용하는 스마트 계약은 작업 증명 솔루션에 비해 상당한 성능 이점을 보장한다. 광범위한 산업에서의 채택을 위해서는 다양하고 중요한 요구사항이 추가로 충족되어야 한다. 예를 들어, 안정적인 하위 호환 스마트 계약 시스템은 간단한 지불 검증 (SPV) 기술을 지원하는 라이트 모바일 지갑으로, 조직간의 정보 - 물류 통합을 자동화해야 한다. 현재 선도적인 스마트 계약 솔루션 인 Ethereum은 계산 비용이 많이 드는 Proof of Work (POW) 를 사용하며 향후 여러 번 하드 포크가 예상되며 전체 블록 체인을 다운로드 해야 한다. 결과적으로, Ethereum 스마트 계약은 제한된 유용성을 가지며 보안상의 문제가 부족하다 이 백서는 사회 기술 적용 적합성, 형식의미론 언어표현력 채택 및 빠른 모범업계사례 배포를 위한 스마트 계약 템플릿 라이브러리의 제공을 목표로 하는 Qtum 스마트 계약 프레임 워크를 제시함으로써 최신 기술을 선보인다. 우리는 Ethereum 대안에 비해 Qtum 유틸리티의 이점에 대해 논의하고 산업 케이스 어플리케이션을 위한 Qtum 스마트 계약 미래 개발 계획을 제시한다.

핵심 용어 : 스마트 계약, 비즈니스 네트워크 모델, DAPP, 모바일, 정보 작업, cross-organizational, peer-to-peer, 분산 시스템, e-governance, Qtum 프레임워크

1. 도입

편의, 검증 및 컴퓨팅 작동을 위한 프로토콜 편성과 구성은 서로 간의 동의 된 당사자간의 협상된 계약을 의미한다. 이를 스마트 계약이라 부른다. 스마트 계약의 경우 초기에는 다양한 분야에서 어플리케이션을 찾았다. 예를 들면 금융 기술[6], IoT(Internet-of-Things) 어플리케이션[33], 디지털 서명 솔루션[11] 등이 있다. 스마트 계약의 핵심은 거래내역을 분권화를 통한 검증작업이다.

초기에는 작업증명(Proof-of-Work)[42]이라는 방법을 통해서 검증작업을 했다. 스마트 계약을 가능하게 한 핵심기술은 신뢰할 수 있는 중앙 기관을 두지 않고 거래내역을 기록하는 블록체인(blockchain)이라 불리는 공개 분산 장부를 사용하는 것이다. 블록체인 기술은 계층화 프로토콜 위에서 제한된 작업 집합으로 구성된 Peer-to-peer(P2P) 암호화 지폐 및 지불 방법인 Bitcoin[23]을 시작으로 널리 보급되었다. Bitcoin은 거래내역 검증을 위해서 연산량이 많고 전기를 많이 사용하는 PoW 방식을 사용한다.

Bitcoin과 달리 많은 스마트 계약 시스템에는 JavaScript 문법과 배포 방식이 유사한 Turing-complete 언어인 Solidity가 탑재되어 있다. 이에 대표적인 자용 예가 이더리움 가상 컴퓨터[44]이다. 이더리움은 몇 가지 결함에도 불구하고 실질적으로 스마트 계약 시스템을 선도하고 있다. 첫 번째로, 거래내역 검증을 위한 작업증명 방식은 이더리움의 대부분의 산업 어플리케이션에 실현 가능하지 않다고 여기는 시점까지 확장성을 감소 시키고 있다. 두 번째로, 최근 크라우드펀드 사례에서 이더리움 계열의 Solidity 스마트 계약은 정식 검증 도구[3]와 비교해서 최신기술이 부족하여 보안상의 결점으로 인하여 해킹을 당했다. 보안의 결점으로 인하여 약 5천만 달러의 손실이 발생했다. 결과적으로 이더리움은 hardfork를 통해서 2개의 이더리움이 생성되었다. 하지만 이더리움의 hardfork의 경우 Denial-of-Service(DoS) 공격으로 인해 발생 했기 때문에 지분증명 방식[2]을 사용한 거래내역 검증과 블록체인 샤딩[20]을 실현시키기 위해서 더 많은 hardrork 가 반드시 필요하다.

이 외에도 더 많은 이유들로 인하여 광범위한 산업에 적용[8]되는 이더리움은 제한적이다. 예를 들어 상호-조직적인 정보-실행계획의 자동화가 불가능하고, 외부와 관련된 내부 전용 계약과의 차별화 된 개인 정보 보호 기능이 없으며, 블록체인을 위한 보다 좋은 성능의 지분증명 방식[2]을 통한 거래내역 검증이 가능한 안전하고 안정적인 가상 머신, 공식적으로 검증이 가능한 스마트 계약 언어, 전체 블록체인을 다운로드 할 필요 없는 경량 지갑 그리고 스마트 계약을 위한 간단 지불 증명(Simple Payment Verification, SPV)[14]을 지원하는 모바일 장치 솔루션이 있다. 블록체인의 경우 고객이 임의의 전체 노드[23]에 연결할 때 단지 블록 헤더만 다운로드 하는 것을 의미합니다. Qtum은 Ethereum Virtual Machine(EVM)을 사용하면서 적합한 대안을 찾지 못하고 있지만[19], [19]에 의하면 EVM 은 이전에 경험한 mishandle 예외에 대한 공격 이라던 가 거래내역 순서, 타임스탬프와 같은 종속성과 같은 결함이 있습니다. 스마트 계약 시스템이 사이드 체인과미 사용 거래 결과(UTXO)[10]를 사용하여 Bitcoin[23]과 색다른 코인[36]과 같은 다른 블록체인 시스템과 호환성을 성공 시켜서 산업 확장성을 달성하는 것이 좋다. 또한 Bitcoin Lightning Network[35]의 기능을 사용하면 양방향 소형지불 수단을 통해서 확장성을 확보 할 수 있다.

이더리움과 같은 스마트 계약 시스템이 주목 받고 있지만 위에서 이미 언급한 이유들로 인하여 광범위한 선업적용이 이루어 지고 있지 않다. 이 백서는 상호-조직적인 정보-실행계획의 비용과 시간을 줄이기 위해 중요한 고객 요구 사항을 충족시키기 위한 스마트 계약 솔루션을 개발하는 방법에 대한 질문에 대답하는 스마트 계약 시스템용 Qtum 프레임워크를 명시함으로써 그 차이를 설명한다. 우려 사항을 구분하기 위해서 다음과 같은 하위 질문을 제기해 본다. Qtum 스마트 계

약 솔루션이 제공하는 차별화 된 기술적 성능 이점은 무엇인가? Qtum 프레임워크가 만족하는 중요한 스마트 계약 요구 사항은 무엇인가? Qtum 프레임워크가 지향하는 상호-조직적인 정보-실행 계획의 고유한 특징은 무엇인가?

이 백서의 나머지 부분은 다음과 같이 구성되어 있다. 첫 번째, 섹션2 에서는 관련 솔루션과 비교하여 기술적인 성능향상을 이루기 위한 Qtum 프레임워크의 구체적인 장점에 중점을 둔다. 섹션3 에서는 사회기술적으로 조직화 된 스마트 계약 시스템에 대한 관련 이해 관계자와 함께 기능 및 품질 목표를 제시한다. 섹션4 에서는 Qtum 프레임워크 가치-전송 프로토콜이 실행중인 상황에 어떻게 제공하는지 보여준다. 마지막으로, 섹션5 에서는 한계, 미해결 문제 및 향후 계발 방향에 대해 논의하면서 이 백서를 마무리 한다.

2. Qtum의 성능 이점

Qtum의 주요 목표 중 하나는 지분 증명(PoS)[37]합의 모델을 사용하여 UTXO기반 스마트 계약 시스템을 구축하는 것이다. 지분 증명 방식은 다음 블록 생성자가 암호화화폐에서 보유한 가치를 기반으로 선택된다는 것을 의미한다. 블록은 일반적으로 채굴되는 대신 위조 혹은 조폐가 되므로 거래 수수료 이외에 블록 보상이 있으며 위조자는 자신이 맡기는 지분에 대한 비율로 "이자"을 지급 받는다.

Qtum은 Bitcoin 및 이더리움 생태계와 호환되며 이더리움 가상 머신(EVM) 호환성을 갖추면서 Bitcoin을 변형하는 것을 목표로 한다. 주목할 만 것은 이더리움과 달리 Qtum EVM은 지속적으로 하위 호환이 가능하다. 실용적인 디자인 접근 방식을 추구하는 Qtum은 모바일 장치로 구성된 전략으로 산업계 사용 사례로 이용한다.산업계 사용 사례는 Qtum이 다양한 인터넷 사용자에게 블록체인 기술을 홍보함으로써 PoS 거래내역 검증을 분산시키는 것을 허용한다.

나머지는 다음과 같이 구성된다. 2.1절에서는 Bitcoin UTXO와 이더리움 계정 모델의 장점을 비교한다. 다음 2.2절에서는 Qtum 블록체인에 대한 합의 플랫폼에 대해 설명한다. 2.3절에서는 Qtum 계약을 EVM에 통합하는 방법을 보여준다. 마지막으로 2.4절에서는 Qtum 작업에 대한 지불 모델을 설명한다.

2.1 UTXO 와 계정 모델

UTXO 모델에서 거래내역은 사용되지 않은 Bitcoin으로 사용 되며 새로운 UTXO가 생성되는 것이 거래내역 출력이 된다. 미사용 거래내역 출력은 변경되면 생성되고 소비자[1]에게 반환된다. 이러한 방식으로 일정량의 Bitcoin은 다른 개인키 소유자 사이에서 전송되고 새 UTXO 거래내역 체인에서 소비되고 생성된다. Bitcoin 거래내역의 UTXO는 수정된 거래내역 버전을 서명하는데 사용되는 개인키로 잠금이 해제된다. Bitcoin 네트워크에서 채굴자는 코인베이스 거래내역이라 불리는 어

떤 기록도 가지고 있지 않은 Bitcoin을 생성한다. Bitcoin은 작업 집합으로 제한된 거래내역에 스크립트 언어를 사용한다. Bitcoin 네트워크상의 스크립트 시스템은 Last-In, Fist-Out의 LIFO원칙에 따라 추상화된 데이터 유형인 스텍별로 (주스텍 과 Alt 스텍) 데이터를 처리한다.

Bitcoin 클라이언트에서 개발자는 `isStandard()`함수[1]를 사용하여 스크립트 유형을 요약한다. Bitcoin 클라이언트 지원 스크립트 유형: P2PKH(Pay to Public Key Hash), P2PK(Pay to Public Key), MultiSignature(15개 미만 개인키 서명), P2SH(Pay to Script Hash), OP_RETURN. 이 5가지 표준 스크립트 유형을 통해 Bitcoin 클라이언트는 복잡한 지불 논리를 처리 할 수 있다. 이 외에도 채굴자가 비표준 거래내역을 캡슐화 하는데 동의하면 비표준 스크립트를 작성하고 실행 할 수 있다. 예를 들면 스크립트 생성 및 실행 프로세스에 P2PKH를 사용하면 가상의 Bitcoin 주소 "빵집 주소"를 가지고 있는 제과점에서 빵에 대해 0.01BTC를 지불하는 것을 가정해보자. 이 거래내역의 출력은 다음과 같다.

OP_DUP OP_HASH160 <Bread Public Key Hash> OP_EQUAL OP_CHECKSIG

OP_DUP 연산은 스택의 최상위 항복을 복제한다. OP_HASH160은 Bitcoin 주소를 최상위 항목으로 반환한다. Bitcoin의 소유권을 설정하려면 디지털 키와 디지털 서명 이외에 Bitcoin 주소가 필요하다.OP_EQUAL 은 상위 두 항목이 정확히 일치하면 TRUE(1)을 반환하고 그렇지 않으면 FALSE(0)을 반환한다. 마지막으로 OP_CHECKSIG는 거래내역의 해쉬 된 데이터와 관련된 서명에 대한 유효성 검사와 함께 공개키와 서명을 생성하고 일치하면 TRUE를 반환한다.

잠금 스크립트에 따른 잠금 해제 스크립트는 다음과 같다.

<Bread Signature><Bread Public Key>

위의 두 스크립트를 결합한 스크립트:

<Bread Signature><Bread Public Key> OP_DUP OP_HASH160

<Bread Public Key Hash> OP_EQUAL OP_CHECKSIG

잠금 해제 스크립트와 잠금 스크립트가 사전 정의된 조건과 일치하는 경우에만 참인 스크립트 조합이 실행된다. Bread Signature 는 유효한 빵집 주소의 개인키와 일치시켜서 서명해야 하며 결과가 참이어야 함을 의미한다.

불행하게도 Bitcoin 스크립트 언어는 Turing-complete 가 아니다.예를 들면 스크립트 언어에서 루프 기능은 없다. 이러한 제한요소는 복잡한 지불 조건의 발생을 방지함으로써 무한 루프 또는 기타 복잡한 논리 허점과 같은 보안 위험을 완화한다.

UTXO 모델에서는 공공 장부를 통해서 각 거래내역을 투명하게 추적할 수 있다. UTXO 모델은 확장 가능한 여러 주소 사이에서 거래내역을 초기화하는 병렬 처리 기능을 가지고 있다. 또한 UTXO 모델은 사용자가 UTXO의 출력으로 주소 변경을 할 수 있다는 점에서 개인 정보 보호를 지원한다. Qtum의 목표는 UTXO 모델의 혁신적인 디자인을 기반으로 스마트 계약을 수행하는 것

이다.

UTXO 모델과 비교하여 이더리움은 계정 기반 시스템이다. 보다 정확하게 각 계정은 상태 변화와 함께 직접적인 가치 및 정보 전달을 수행한다. 이더리움 계정 주소 20 바이트는 거래내역에 대한 일회성 처리, Ether라 불리는 거래 수수료 지불을 위한 주 내부 암호 연료의 잔고, 선택적 계약 코드 및 기본 공계정 저장소를 보장하기 위한 카운터로 구성된다.

두 종류의 Ether 계정은 한편으로 개인 키로 제어 되고 또 다른 한편으로는 계약 코드가 제어 된다.이전 무효-코드 유형은 메시지 전송을 위한 거래내역을 생성하고 서명한다.이는 내부 저장소 읽기 및 쓰기, 계약 생성 또는 다른 메시지 전송에 대한 메시지를 받은 이후 코드를 활성화한다.

이더리움의 경우 잔액 관리가 실제 은행 계좌 관리와 비슷하다. 새로 생성된 모든 블록은 잠재적으로 다른 계정의 글로벌 상태에 영향을 준다.모든 계정에는 다른 계정 또는 주소를 호출 할 수 있는 자신만의 잔고, 코드-공간이 있고 각각의 실행 결과가 저장된다. 기존 이더리움 계정 시스템에서 사용자는 클라이언트 원격 절차 호출을 통해서 P2P 거래내역을 수행한다.스마트 계약을 통해 더 많은 계정으로 메시지를 보낼 수 있지만 이러한 내부 거래는 각 계정의 잔고에서만 볼 수 있으며 이더리움의 공개 장부에서 이를 추적하기 어렵다.

지금까지 논의된 내용을 기반으로 이더리움 계정 모델이 확장성 병목으로 간주하고 Bitcoin-network UTXO 모델의 장점이 명확해 진다.**이는 우리가 원하는 네트워크 효과를 향상 시키기 때문에 Qtum 배포가 보류가 된 사항에 대해 UTXO 모델의 채택은 디자인 결정에 필수적인 요소이다.**

2.2 합의 관리

합의와 플랫폼이 각 프로젝트 요구사항을 충족시키는지에 대한 논의가 진행중이다. 가장 널리 논의되는 주제는 다음과 같다: PoW[41], PoS[2], Dynamic PoS, HyperLeger에서 논의된 비잔티움 오류 범위오차. 합의의 본질은 분산된 알고리즘으로 데이터 일관성을 이루는 것이다.가능한 옵션은 다음과 같다. 노드 사이에서 100% 합의가 없으면 합의에 도달 할 수 없다는 Fischer Lynch 와 Paterson 의 이론[5].

Bitcoin 네트워크에서 채굴자는 PoW방식의 해쉬 출동에 의한 검증 과정을 통해 참여한다.채굴자의 해시 값을 계산할 수 있고 아래의 조건을 충족할 경우 채굴자는 새로운 블록을 채굴했다고 네트워크상에 주장 할 수 있다:

$$Hash(BlockHeader) \leq \frac{M}{D}$$

채굴자 M과 채굴 난이도 D의 경우 `Hash()`는 SHA256의 값의 범위 [0,M] 과 D를 의미한다. Bitcoin에서 사용하는 SHA256 알고리즘을 사용하면 채굴난이도 대비 채굴자 수가 많아질 경우 모든 노드에서 각 블록을 재빠르게 확인 할 수 있다. 80바이트 BlockHeader는 각각의 Nonce에

따라 달라진다.채굴의 전반적인 난이도는 블록체인 네트워크의 전체 해쉬 파워에 따라 동적으로 조정된다. 두 명 이상의 채굴자가 동시에 블록을 계산하면 small fork가 네트워크상에서 발생한다. 이것은 블록체인이 받아들여야 할지 거부해야 할지 결정을 위해 잠금 위한 지점이다. Bitcoin 네트워크에서 가장 검증 된 작업을 수행하는 체인이 합법적이다.

대부분의 PoS 블록체인은 그들의 유산을 이전 버전의 Bitcoin Core를 기반으로 하는 PeerCoin으로 되돌릴 수 있다. Scrypt, X11, Groestl, Equihash[4]등과 같은 다양한 PoW 알고리즘이 있다.새로운 알고리즘을 사용하는 목적은 하나의 개체에 의한 컴퓨팅 성능의 누적을 방지하고 Application Specific Integrated Circuits(ASIC)이 도입 될 수 없도록 보장하는 것이다. Qtum Core는 기본 합의형을 위해 최신 Bitcoin 소스 코드를 기반으로 PoS 를 사용한다.종래의 PoS 거래내역에서 새블록 생성은 다음 조건을 충족시켜야 한다:

$$ProofHash < coins \times age \times target$$

ProofHash에서 지분 수정자[40]는 사용되지 않은 출력과 현재 시간을 함께 계산 한다.이 방법을 사용하면 한 명의 악의적인 공격자가 대량의 coin age 에 이중 지출 공격을 시작 할 수 있다. Coin age 에 의해 제기되는 또 다른 문제점은 노드가 보상을 받은 후 지속적으로 온라인 상태가 아닌 간헐적인 온라인 상태에 있다는 것이다.따라서 PoS 동의가 개선된 버전에서는 coin age를 삭제함으로써 더 많은 노드가 동시에 온라인 상태가 되도록 한다.

원래의 PoS 구현은 가능한 coin age 공격 및 다른 유형의 공격[16]으로 인해 여러가지 보한 문제로 어려움을 겪는다. Qtum은 Blackcoin[40]팀의 보안 분석에 의견을 일치시켜 PoS 3.0을 최신 Qtum Core에 채택한다. PoS 3.0은 이론적으로 지갑을 오프라인 상태로 유지하는 코인 보유자에게는 인센티브를 주지 않고 코인을 더 오래 연결 시키는 투자자에게 보상한다.

2.3 Qtum 계약과 EVM 통합

EVM 은 스택 기반 256-비트 머신 단어이다. 이더리움에서 실행되는 스마트 계약은 집행을 위해 이 가상 시스템을 사용한다. EVM은 이더리움의 블록체인을 위해 설계되었므로 모든 가치 전송이 계정 기반 방법을 가정한다. Qtum은 Bitcoin의 블록체인 디자인을 기반으로 하며 UTXO 기반 모델을 사용한다. 따라서 Qtum에는 UTXO 기반 모델을 EVM의 계정 기반 인터페이스로 변환하는 계정 추상화 계층이 있다.유의할 점은컴퓨팅의 추상화 계층은 특정 기능의 세부 구현 사항을 숨겨서 상호 운용성 및 플랫폼 독립성 촉진과 같은 관심사를 구분할 수 있는 수단이 된다.

EVM 통합: Qtum의 모든 거래내역은 Bitcoin과 마찬가지로 스크립트 언어를 사용한다.그러나 Qtum에는 3개의 새로운 opcode가 있다.

- OP_EXEC: 이 opcode는 거래내역(아래에 설명 됨)의 특수 처리를 유발하고 특정 입력 EVM 바이트 코드를 실행한다.

- OP_EXEC_ASSIGN: 이 opcode 역시 OP_EXEC와 같은 특수 처리를 유발한다.이 opcode에는 계약서의 계약 주소와 데이터가 입력된다. 다음은 주어진 데이터를 전달하면서 계약 바이트 코드의 실행을 따른다(EVM에서 CALLERDATA로 주어진다).이 opcode는 선택적으로 스마트 계약에 돈을 송금한다.

- OP_TXHASH: 이 opcode는 계정 추상화 계층의 홀수 부분을 조정하고 현재 실행된 거래내역의 ID 해쉬를 푸쉬 하는데 사용된다.

통상적으로 스크립트는 출력을 소비할 때만 실행된다.예를 들어, 스크립트가 표준 공개 키 해쉬 거래내역을 사용하여 블록체인에 있는 동안 유효성 검사 또는 실행이 수행되지 않는다.거래내역 입력이 출력을 참조 할 때까지 실행 및 유효성 검사가 수행되지 않는다.이 시점에서 거래내역 입력 스크립트(ScriptSig)가 출력 스크립트에 유효한 데이터를 제공하여 후자가 0이 아닌 값을 반환하는 경우에만 유효하다.그러나 Qtum은 블록체인에 병합 되는 즉시 실행되는 스마트 계약을 충족시킨다.그림 1에서와 같이 Qtum은 OP_EXEC또는 OP_EXEC_ASSIGN을 포함하는 거래내역 출력 스크립트(ScriptPubKey)의 특수 처리를 통해 이 작업을 수행한다.이러한 opcode중 하나가 스크립트에서 감지되면 거래내역이 블록에 배치 된 후 네트워크의 모든 노드에 의해 실행된다.이 모드에서 실제 Bitcoin 스크립트 언어는 스크립트 언어로 사용되는 것이 아닌 EVM으로 데이터를 전송한다.이는 이더리움 계약과 마찬가지로 opcode중 하나에 의해 실행 될 때 자체 상태 데이터베이스 내에서 상태를 변경한다.

Qtum 스마트 계약을 쉽게 사용하려면 스마트 계약뿐 아니라 특정 pubkeyhash주소에서 파생된 작성자에게 전송된 데이터를 인증해야 한다. Qtum 블록체인의 UTXO 집합이 너무 커지지 않도록 OP_EXEC및 OP_EXEC_ASSIGN거래내역 출력도 사용 할 수 있다.OP_EXEC_ASSIGN출력은 계약서에서 코드가 다른 계약서나 pubkeyhash주소로 돈을 보낼 때 사용된다.OP_EXEC출력은 계약서에서 자기 제거 작업을 사용하여 블록체인 자체를 제거 할 때 마다 소비된다.

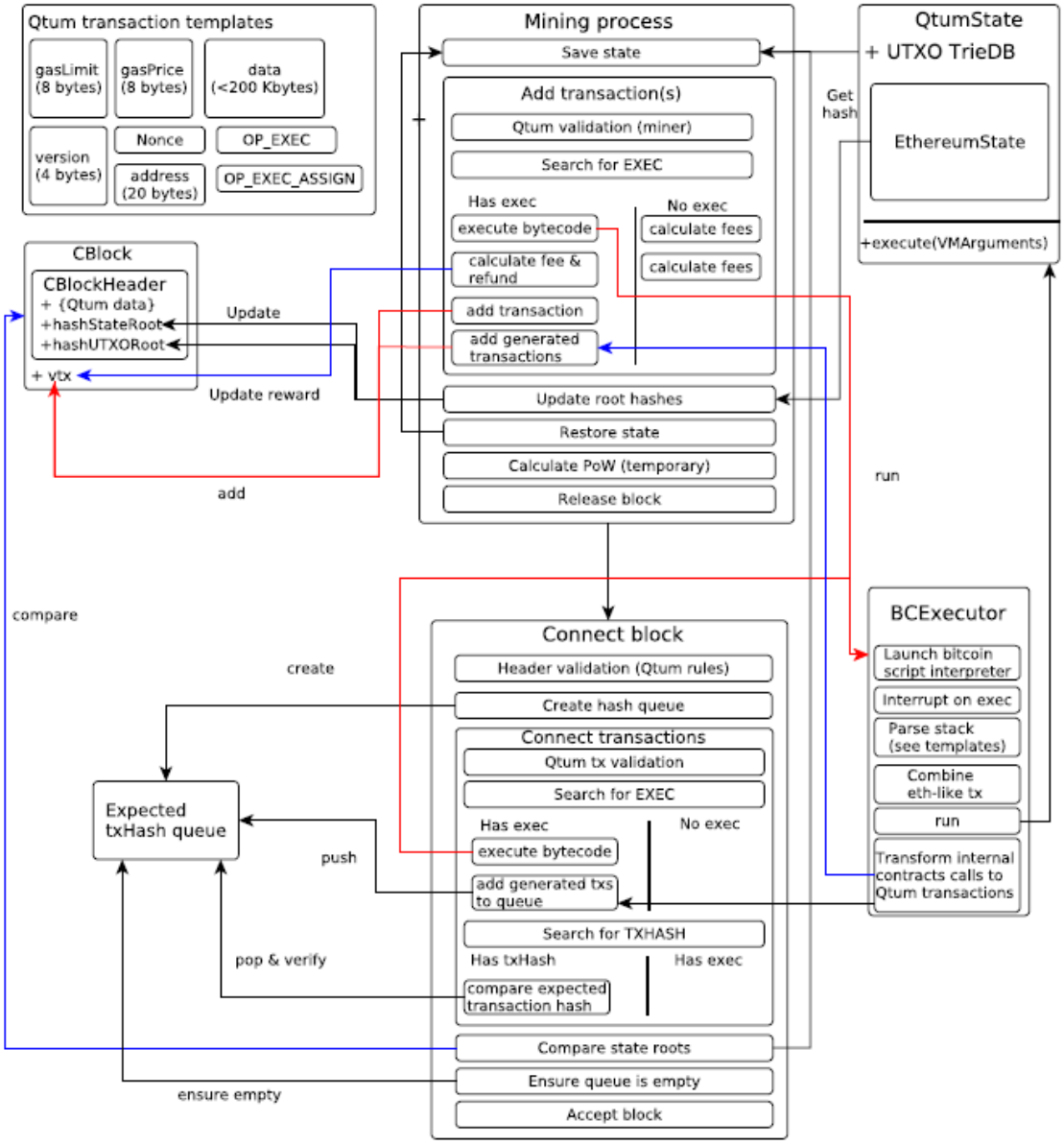

그림1. Qtum 거래내역 과정

Qtum 계정 추상화 계층EVM은 계정 기반 블록체인에서 작동하도록 설계되었다.그러나 Qtum은 Bitcoin을 기반으로 UTXO 기반 블록체인을 사용하면 가상 컴퓨터 및 기존 이더리움 계약을 크게 수정하지 않고 EVM을 Qtum 블록체인에서 작동 할 수 있게 해주는 계정 추상화 계층(AAL)을 포함한다.

EVM 계정 모델은 스마트 계약 프로그래머에게 사용하기 쉽다.블록체인에서 현재 계약과 기타 계약의 잔액을 확인하는 작업이 있으며 데이터(데이터 첨부)를 다른 계약으로 보내는 작업이 있다. 이러한 동작이 상당히 기본적이고 최소한으로 보일지라도 UTXO 기반 Qtum 블록체인 내에서 적용하기는 란 쉽지 않다.따라서 이러한 작업내의 AAL 구현은 예상 보다 복잡할 수 있다.

스마트 계약이 배포된 Qtum 블록체인은 해당 주소로 할당 되고 호출 가능하며 새로 배치 된 계약 잔액은 0으로 설정 된다.현재 Qtum에는 잔액이 0이 아닌 계약을 배포할 수 있는 프로토콜이 없다.계약에 자금을 보내기 위해서 거래내역은 OP_EXEC_ASSIGN opcode를 사용한다.

아래 예제 스크립트는 돈을 계약서에 보낸다:

```
1; VM 버전

10000; 거래내역을위한가스한계

100; Qtum 사토시단위가스가격

0xF012; 계약서에보내는데이터(일반적으로 Solidity ABI를사용함)

0x145222265803b201ac1f8bb25840cb70afe3303;

거래내역 txid의 Ripedmd-160 해쉬함수

OP_EXEC_ASSIGN
```

위의 간단한 스크립트는 거래내역 처리를 통해 OP_EXEC_ASSIGNopcode로 넘긴다.Out-of-gas가 발생하지 않았거나 기타 예외가 발생하면 계약에 부여 된 금액이 OutputValue 이다.우리가 논의하는 가스 메커니즘의 정확한 세부 사항은 아래에 있다.이 출력을 블록체인에 추가함으로써 출력이 소유한 계약서의 도메인로 들어간다.이 출력 값은 예약의 잔액에 지출 가능한 출력의 합계로 반영된다.

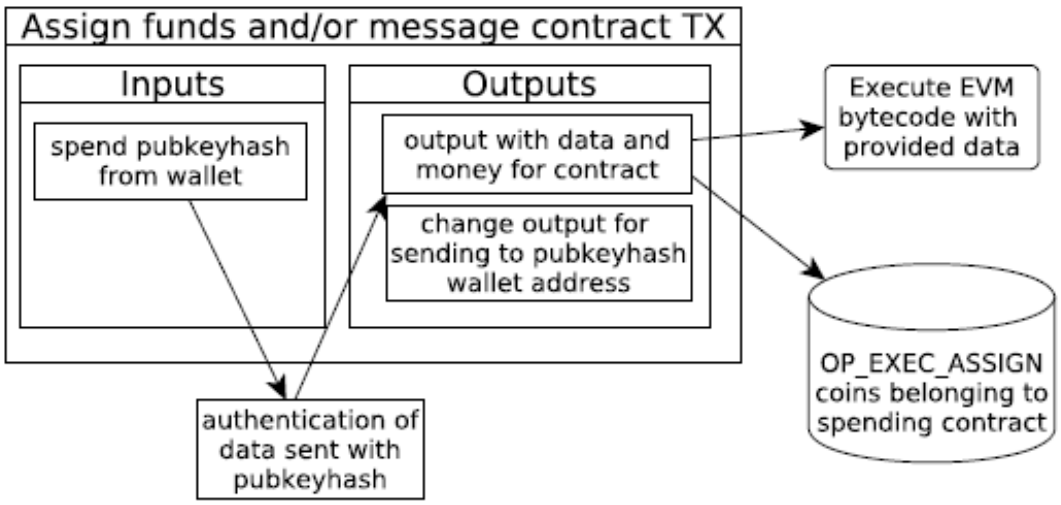

그림2. 기금 및/또는 미시지 계약서 할당

그림 2는 표준 공개키 해쉬 출력에서 계약에 자금을 보내는 것을 보여주지만 하나의 계약에서 다른 계약으로 돈을 보내는 방법은 거의 동일 하다.계약이 다른 계약 또는 공개키 해쉬 주소로 자금을 보내는 경우 전자는 소유한 출력 중 하나로 보낸다.송금 계약에는 자금 송금에 대한 예상 계약 거래가 포함된다.이러한 거래내역은 Qtum 네트워크에 유효한 블록에 있어야 한다는 점에서 특별하다.예상 계약 거래내역은 소비자가 생성하는 것이 아니라 거래내역을 확인하면서 채굴자가 생성한다.따라서 P2P 네트워크에 반포되지 않는다.

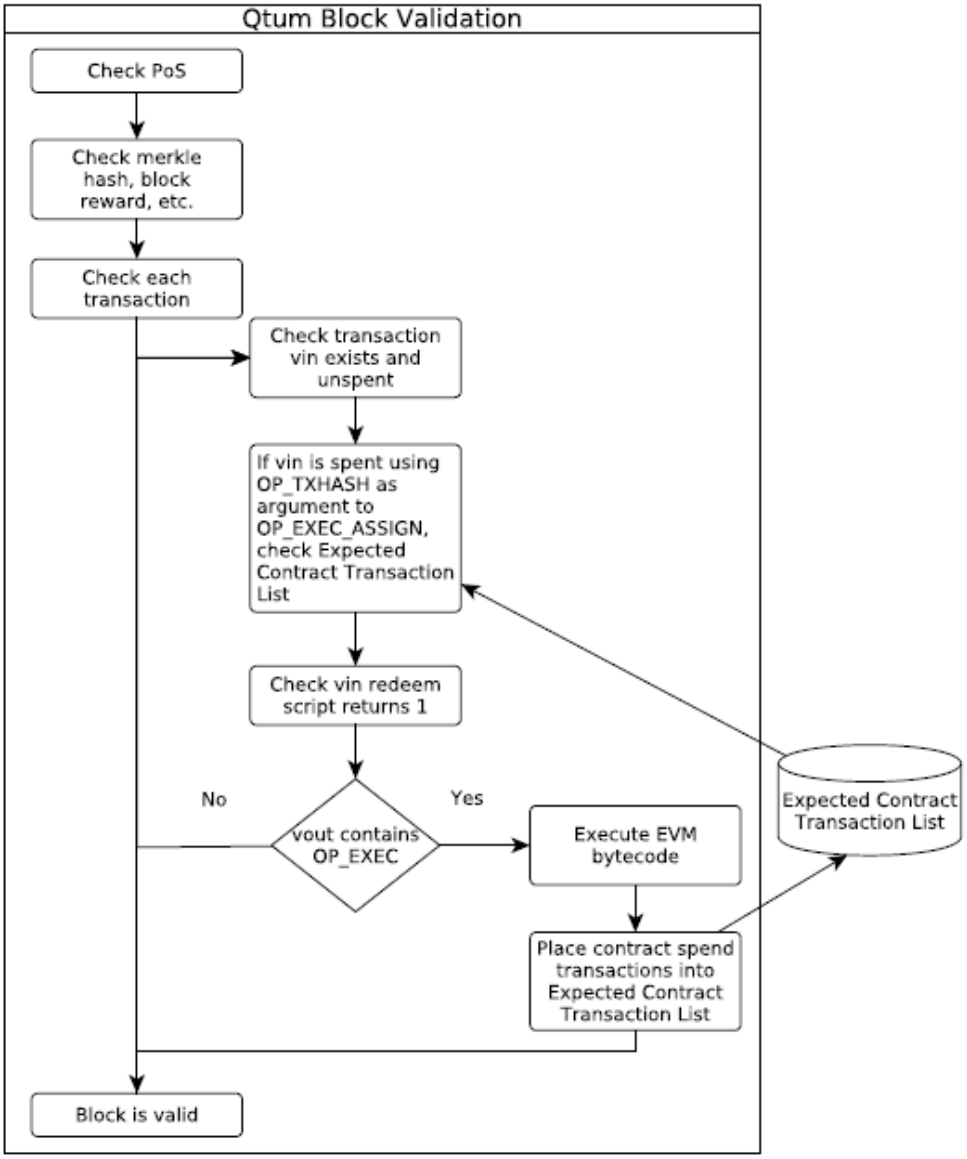

그림 3.예상 계약 거래내역 목록을 보여주는 Qtum 블록 유효성 검사

예상 계약 거래내역을 수행하는 기본 메커니즘은 그림 3의 일부인 새로운 opcode인

OP_TXHASH이다.내부적으로 OP_EXEC와 OP_EXEC_ASSIGN에는 두가지 모드가 있다.일정 부분의 출력 스크립트가 실행되면 EVM이 실행된다.그러나 입력 스크립트 처리의 일부로 opcode가 실행되면 EVM은 이중 실행을 피하기 위해 실행되지 않는다.그 대신 OP_EXEC 및 OP_EXEC_ASSIGN opcode는 no-ops와 유사하게 행동하고, 주어진 거래내역 해쉬에 기초하여 각각 1 혹은 0, 즉 소비 가능 소비 불가능으로 반환한다.이점이 OP_TXHASH가 기능 개념의 가장 중요한 이유이다.간단하게 OP_TXHASH는 현재 소비 거래내역의 SHA256 해쉬를 Bitcoin 스크립트 스택에 푸쉬하는 새로운 opcode이다. OP_EXEC및 OP_EXEC_ASSIGN opcode는 지출 과정에서 예상 계약 거래 목록을 확인한다.

거래내역이 예상 계약 거래내역 목록에 있는 opcode로 전달 된 후(일반적으로 OP_TXHASH에서 나온) 결과는 1 혹은 소비가능 이다. 이러한 방식으로 vout을 사용하는 OP_EXEC 및 OP_EXEC_ASSIGN은 계약서에서 계정 추상화 계층에서 vout을 사용할 수 있도록 요구 할 때만 사용 할 수 있다(예: 계약 송금 시도 과정). 그 결과 정상적인 UTXO 거래내역과 일치하는 계약에서 계약 자금만 소비되도록 안전하고 건전한 방법으로 진행 된다.

계약서에 지출 할 수 있는 출력이 두개 이상 있는 경우 특정 시나리오가 생성된다. 각 노드는 서로 다른 출력을 선택할 수 있기 때문에 OP_EXEC_ASSIGN거래내역을 사용하기 위해 완전히 다른 거래내역을 사용한다. 이는 Qtum 에서 핵심-합의 코인 선택 알고리즘에 의해 해결된다.이것은 사용자 지갑 내에서 사용되는 표준 코인 선택 알고리즘과 유사하다.그러나 Qtum은 알고리즘을 간소화하여 Denial-of-Service(DoS)공격의 위험을 피하고 간단한 합의 규칙을 실현한다. 이런 핵심-합의 코인 선택 알고리즘을 사용하면 다른 노드가 다른 코인을 선택하여 계약에의 소비될 가능성이 없다.다른 출력을 선택하는 모든 채굴자/노드는 주 Qtum 네트워크에서 분기되어야 하며 블록은 유효하지 않게 투영된다.

그림 4의 EVM 계약이 pubkeyhash주소 혹은 다른 계약으로 돈을 보내면 이 이벤트는 새 거래내역을 구성한다.핵심-합의 코인 선택 알고리즘은 가장 좋은 결과를 소유한 계약 풀을 선택한다.이런 출력은 단일 OP_TXHASH연산 코드를 구성하는 입력 스크립트(ScriptSig)와 함께 입력으로 사용된다.따라서 출력은 거래내역의 나머지 잔액을 계약서로 보내는 변경된 출력(만약 필요한경우)이다.이 거래내역 해쉬는 예상 계약 거래내역 목록에 추가 된 다음 거래내역 자체가 계약 거래내역이 실행된 직후 블록에 추가된다.이렇게 생성된 거래내역이 확인 되고 실행되면 예상 계약 거래내역 목록 확인 수행된다.그런 다음 이 거래내역 해쉬는 예상 계약 거래내역 목록에서 제거된다.이 모델을 사용하면 OP_TXHASH를 사용하는 대신 하드 코딩 된 해쉬를 입력 스크립트로 제공하여 거래내역을 위장하는 것은 불가능하다.

위에서 설명한 추상화 계층은 EVM 계약을 코인 선택 및 특정 출력에 사용하지 않는다.대신 EVM 계약서는 자신과 다른 계약이 잔고만을 알고 있기 때문에 이런 계약서뿐만 아니라 계약 시스템 밖의 pubckeyhash주소로 돈을 보낼 수 있다.결과적으로 Qtum과 이더리움 간의 강한 계약 호환성으로 인해 이더리움 계약을 Qtum 블록체인에 이식하는데 필요한 수정 사항이 거의 없다.

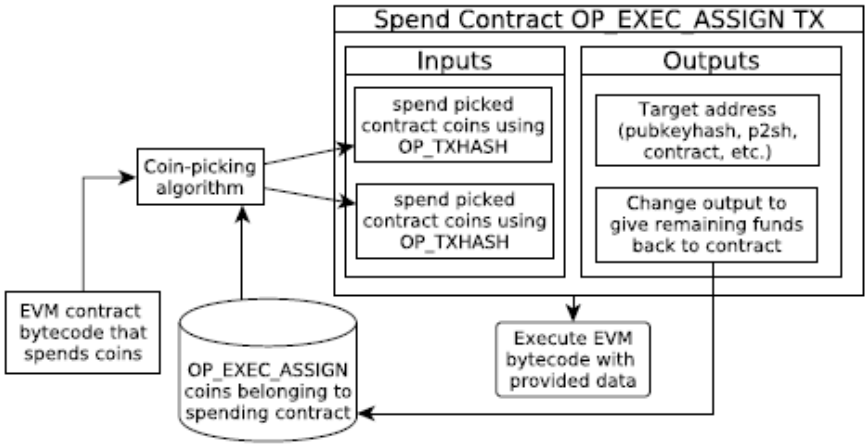

그림 4. OP_EXEC_ASSIGN거래내역 계약서 소비

표준 거래내역 유형 추가: 다음은 Qtum에 추가하는 표준 거래내역 유형이다.여기에 Bitcoin 스크립트 템플릿으로 설명 되어 있다: 새로운 계약을 블록체인에 배포하려면 다음과 같은 출력 스크립트가 필요하다.

```
1; VM 버전
[Gas limit]
[Gas price]
[Contract EVM bytecode]
OP_EXEC
```

블록체인에 이미 배치 된 계약에 자금을 보내려면 아래의 스크립트가 필요하다.

```
1; VM 버전
[Gas limit]
[Gas price]
[Data to send to the contract]
[tip-emd160 hash of contract transaction id]
OP_EXEC_ASSIGN
```

예상 계약 거래 목록을 필요로 하므로 지출에 대한 표준 거래 유형이 없다.따라서 이러한 지출 거래내역은 P2P 네트워크에 반포되지 않거나 유효 하지 않다.

2.4 Gas 모델

Qtum이 Bitcoin 블록체인에 turing-completeness 를 추가 할 때 직면하게 되는 문제는 거래의 규모에만 의존하기 때문에 채굴자에게 지불하는 수수료를 결정하기에 적합하지 않다.그 이유는 거래내역 처리를 하는 채굴자가 전체 블록체인을 부한 루프로 돌리고 정지 시킬 수 있기 때문이다. 그림 5에서 볼 수 있듯이 Qtum 프로젝트는 이더리움의 가스 개념을 채택한다. 가스 개념에서 실행 된 각 EVM opcode에는 가격이 있으며 각 거래에는 지출 할 가스의 양이 있다.거래 이후 잔여 가스는 발송인에게 환불된다.

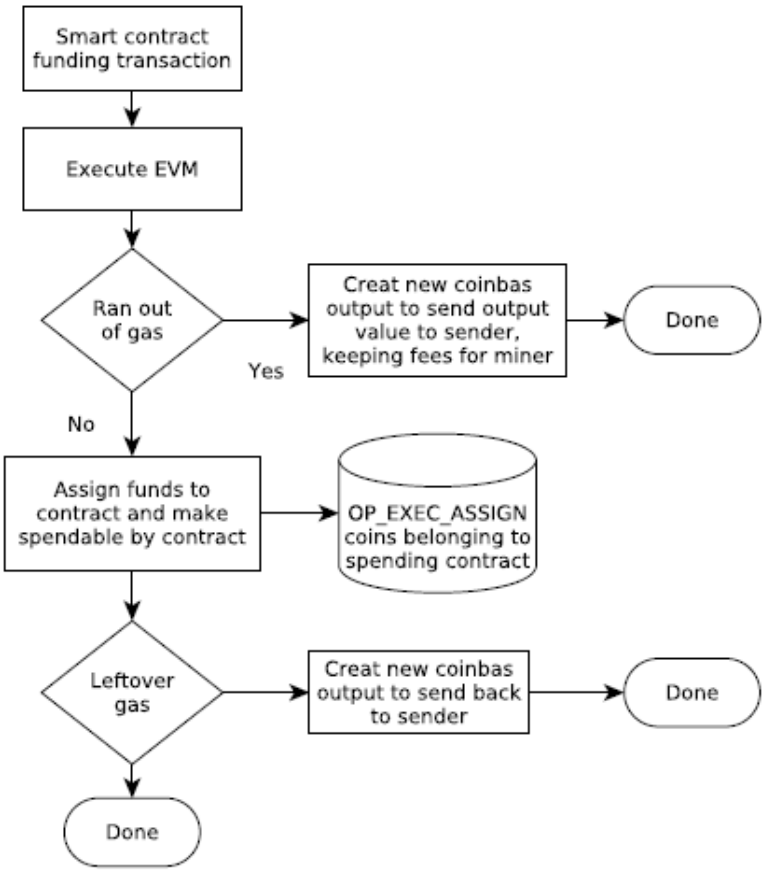

그림 5. 가스 환불 모델

계약 이행에 필요한 가스가 거래에 사용 가능한 양을 초과하면 거래의 행동과 변경된 상태는 되돌려진다.따라서 수정 된 영구 저장 장치는 계약 자금의 지출을 포함 하여 원래의 상태로 되돌아가고 후자는 지출 되지 않는다.원상 복구에도 불구하고 컴퓨팅 리소스가 이미 소비되었기 때문에 거래내역의 모든 가스는 소비되고 처리해주는 채굴자에게 제공된다. Qtum은 이더리움의 가스모

델을 사용함에도 불구하고 가스 일정, 즉 각각의 EVM opcode의 가스 가격이 이더리움과 크게 다를 것으로 예상한다.정확한 값은 이더리움의 기존 가격을 Qtum에 대해 각 opcode에 필요한 과정 및 블록체인 리소스의 양에 의해 결정된다.

계약 자금 조달 또는 배치된 거래내역을 생성 할 때 사용자는 가스에 대해 두개의 특정 항목을 지정한다. `GasLimit`은 계약 실행에 따른 소모 가스량을 결정한다.두번째 항목은 Qtum 사토시에서 각 가스 단위의 정확한 가격을 결정하는 `GasPrice`이다.이는 현재 블록체인이 기록하는 Bitcoin 통화보다 더 작은 단위이다.계약 실행의 최대 Qtum 지출은 `GasLimt`과 `GasPrice`를 곱한 값과 같다.이 최대 지출액이 거래에서 제공 한 거래 수수료를 초과하면 후자는 유효하지 않으며 채굴되거나 처리 될 수 없다.이 최대 지출을 뺀 나머지 거래 수수료는 거래내역 크기 요금이며 표준 Bitcoin 요금 모델과 유사하다.

거래의 적절한 우선 순위를 결정하기 위해서 채굴자는 두 가지 변수를 고려한다.첫째, 거래 규모 수수료는 거래의 총 크기와 일치해야 한다. 이는 일반적으로 킬로바이트 당 최소 코인 금액에 의해 결정된다. 두번째 변수는 계약 실행의 `GasPrice`이다. PoS 채굴자는 블록에 가장 중요하고 수익성 있는 거래내역을 조합해서 선택한다. 따라서 채굴자와 사용자가 거래 속도와 지불하고자 하는 가격에 가장 적합한 최적의 요금으로 최적화 된 자유시장 요금 모델이 존재한다.

환불:UTXO 모델을 사용하면 거래 수수료로 채굴자에게 송금되는 자금은 협상대상이 아니다.채굴자가 예상보다 처리하기가 쉬운 경우 채굴자가 수수료를 부분적으로 환불하는 것은 불가능하다. 그럼에도 불구하고 가스 모델이 유용하기 위해서는 송금인에게 환급금을 돌려주는 방법이 있어야 한다. 또한 가스가 부족한 거래 상태를 롤백하여 가스 요금을 채굴자에게 반환 할 수 있어야 한다. Qtum에서 가스 요금을 환급하는 것은 채굴자의 코인베이스 거래내역의 일부로 새로운 산출물을 생성함으로써 가능하다.코인베이스 거래내역에 환급 출력이 있어야 함을 보장하기 위해 해로운 블록 검증 합의 규칙을 추가한다.그렇지않으면 채굴자가 가스를 환불하지 않는 것을 선택할 수 있다. 환불은 출력 스크립트를 복사하여 거래 자금 송금자에게 되돌려준다.보안상의 이유로 현재 표준 pubkeyhash 지급 또는 scripthash 지급 스크립트를 사용한다.추가 보안 연구를 통해 추가적으로 제한을 해제 할 계획이다.참고로 OP_EXEC_ASSIGN은 계약 자금을 할당하기 위해 다음과 같은 형식을 갖는다:

입력: (푸쉬 순서)

- 지출을 위한 거래내역 해쉬 [선택 사항]
- 버전 번호 (사용할 VM 버전, 현재는 1개)
- 가스 한도 (실행할 때 사용할 수 있는 최대 가스량)
- 가스 가격(각 가스 단위의 qtum 가격)
- 데이터(스마트 계약에 전달할 데이터)
- 스마트 계약 주소

결과: (팝 순서)

- 지출 가능 여부 (현재 지출이 많은 경우)

결과적으로 아래에 EXEC_ASSIGN예제가 나와 있다:

```
1
10000
100
0xABCD1234...
3d655b14393b55a4dec8ba043bb286afa96af485
EXEC_ASSIGN
```

VM 실행으로 인해 out-of-gas예외가 발생하면 vout 은 보완 스크립트 OP_TXHASH를 사용하여 블록의 다음 거래내역에서 소비된다.

거래내역을 위해 생성된 vout은 vin[0].prevout 스크립트에서 나온 pubkeyhash 스크립트이다.초기 버전의 Qtum에서는 pubkeyhash 보낸 사람만 VM 자금 지원 거래내역을 허용한다. VM을 실행하기 위해 다른 형식을 수용할수 있음에도 EVM의 msg.sender 는 "0" 이며 out-of-gas 혹은 gas-refund 를 요구하면 계약서에 자금이 보관된다.

부분 환불 모델:가스 모델과 관련하여 여러가지 이유로 사용되지 않은 부분도 환불 해야만 한다. 한편으로는 사용자가 계약을 제대로 이행 할 수 있도록 많은 자금을 사용 할 수 있다.여전히 미사용 가스는 Qtum 환불로 반환된다.

가스 반환 값은 전송 거래내역의 vin[0].prevout 스크립트로 블록체인에 표시된다.가스는 표준 Bitcoin 거래 수수료 메커니즘을 사용하여 계약으로 보내진다.따라서 새로운 수수료 모델은 거래 수수료를 내기 위해 약간 증가한다.

```
gas_fee = gas_limit * gas_price
txfee = vin – vout
tx_relay_fee = txfee – gas_fee
refurd = gas_fee – used_gas
```

채굴자가 거래 우선 순위를 결정하기 위해 단일 "신용 가격"값으로 tx_relay_fee와 gas_price를 모두 평가할 수 있도록하는 제안서가 있다.

계약 체결 기간 동안 총 요금에서 가스 토큰을 뺀다.이는 gas_price값을 곱한 값이다.계약 실행을 완료 한 이후,채굴자가 블록 보상을 검색하기 위해 사용하는 코인베이스 거래내역에 출력을 추가 하여 gas_fee의 나머지 부분을 주어진 가스 반환 스크립트로 반환해야 한다. Vin 에 추가된 coinbase는 vin[0].prevout의pubkeyhash 이다.가스 환불을 받으려면 반드시 사용 된

pubkeyhash vout상태이어야 한다.그렇지 않은 경우 가스 환불은 채굴자와 함께 out-of-gas 상태로 유지 되며 송금된 금액은 계약과 함께 유지된다.

현재 거래내역당 하나의 EMV 계약만 실행 할수 있음을 알고 있어야 한다.따라서 두 계약 체결 실행이 거래내역 수수료를 공유하려고 시도하는 케이스는 발생하지 않는다.이 시나리오는 거래내역 당 여러 EVM 실행으로 기존 문제를 해결 한 이후에 사용 할 수 있다.현재 디자인은 거래내역 당 여러 계약 실행을 지원한다.

중요한 GAS 테두리 케이스:채굴자들은 계약 가스 및 펀드 반환 스크립트에 신중해야 한다.이 경우 스크립트 출력으로 인해 블록이 최대 크기를 초과 하면 계약 거래내역을 블록에 넣을 수 없다. 대신,다음 반환 블록에서 가스 반환 스크립트 실행을다시 수행해야 한다.채굴자는 계약을 실행하기 전에 가스 반환 스크립트에 대한 후보 블록에 충분한 용량이 있는지 확인해야 한다.만약 반환할 가스 자금이없으면 환불을 위한 vout 요구사항은 없다.

거래 수수료에는 gas_fee가 포함되는 것이 중요하다.거래내역을 블록에 추가하면 마이너스의 가스 환불이 발생하거나 gas_fee가 거래 수수료보다 낮은 경우 거래내역은 유효하지 않다.

두개 이상의 OP_EXEC또는 OP_EXEC_ASSIGNopcode가 있는 거래내역 출력 스크립트는 유효하지 않다.스크립팅 기능이 제한되지만 재귀- 및 다중-실행 문제가 발생할 가능성이 있다.결과적으로,스크립트가 유효하지 않은지를 결정하기 위해 정적 분석만으로 충분하다.

블록체인 중심의 Qtum 기술 이후 개념적으로 스마트 계약 수명주기 관리를 설명한다.후속편의 개념적 발표는 논문 데이터 베이스 [12, 13,18, 24, 26,27, 32]에 뒷받침 된다는 점에 유의해야 한다.

3. 스마트-계약 관리

위에 명시된 바와 같이우리는 수명주기 관리가 잠재적으로 협업 당사자의 적절한 심사가 베팅하기 전에 발생한다는 점에서 스마트 계약을 확보하는데 필수적이라 가정한다. 우리는 실패한 해산물 배달에서 현실적인 사례를 생각 해 볼 수 있다. 여기에는 비즈니스 거래내역 충돌이 기존 미명세 계약에서 나온다. EU 기업(구매자)은 남아시아 회사(판매자)로부터 12,920kg 의 오징어를 주문한다. 기존 계약에서는 운송인이 물품을 획득 할 때까지 제품 품질의 책임은 판매자에게 있다. 세부 규정은 기존 계약서에 명시되지 않은 상품의 품질에 관련되며 구매자는 운송회사(운반책)로 양도하기 전에 상품을 확인하지 않는다.

스마트 계약 대안은 기존 계약에 존재하는 차등 분쟁을 해결한다. 따라서 3.1절에서는 제시된 Qtum 프레임워크 목표 모델은[18, 26, 27, 32]에서 완전히 공식화된 영리 계약 주기의 속성을 반영한다.

3.1 수명주기-관리 목표

목표를 논의하기 위해 다음과 같은 접근 방식을 사용한다. Agent-Oriented Modeling(AOM) 방법[38]은 조직에 속한 사람이 기술을 사용해서 문제를 해결하기 위해 공동 작업을 수행한다는 것을 고려한 사회 기술적 요구 공학 접근 방식이다. 이 섹션에서는 AOM 목표 모델 유형을 사용하여 실행중인 사례를 지원하는 Qtum 스마트 계약 시스템에 대한 중요한 사회 기술적인 행동 특성을 수집한다. 목표 모델은 문제 영역에 대한 이해를 높이기 위해 기술 및 비기술적 이해 관계자 사이의 의사 소통을 향상 시킨다. AOM 목표 모델은 또한 새로우면서 기민한 소프트웨어 개발 기술을 위한 수단이 된다[39].

그림 6. AOM 목표 모델을 위한 모델링 요소

목표 모델은 그림 6과 같이 세가지 주요 요소로 구성된다. 우리가 목표라고 부르는 기능적 요구사항은 평행사변형이고, 남성으로 묘사 된 역할 그리고 비기능적 요구사항으로 묘사된다. 후자의 경우 구름과 같이 묘사된 소프트웨어 관련 비기능 요구사항의 품질 목표와 타원으로 묘사 된 인간 관련 감정적 목표의 두 가지 변형이 있다. 목표 모델은 무결성 정리가 아닌 중심 루트 값 정리로부터 시작한다. 결과적으로, 가치 명제는 트리 계층 구조에서 각 하위 목표가 상위 목표를 달성하기 위한 측면을 표현하는 하위 목표로 분해 되며 하위 목표는 무결성이어야 한다.목표는 하위 목표에 상속된 역할, 양질 목표 및 감정적 목표를 할당한다.

Qtum 프레임워크의 가치 제안: 우리는 그림 7에서 묘사한 Qtum 프레임워크의 근본 목표를 묘사했고 상호 조직간 정보 및 가치 전송 물류 자동화의 가치 제안이다. 우리는 복잡한 가치 제안을 스마트 계약 수명주기 관리 목표로 분할했다[26, 27, 32]. 다시 말해 *setup, rollout, enactment, rollback, termination*으로 분할된다. 3.2절에서는 더 자세하게 설명하는 정의된 목표가 있다.

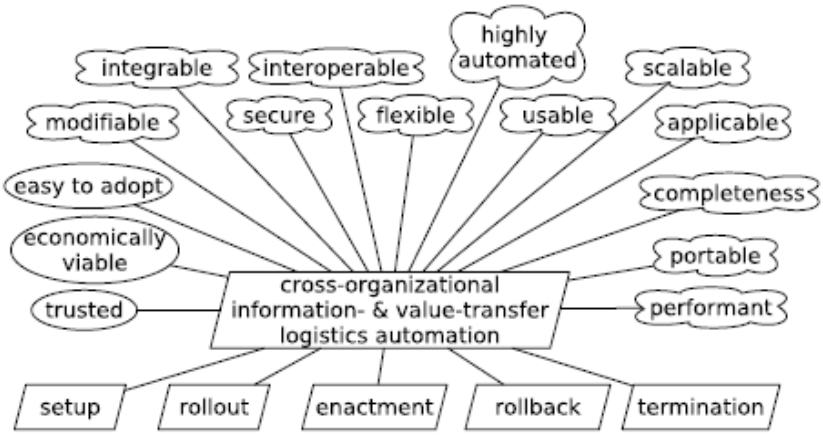

그림 7. Qtum 가치 제안과 수명주기 관리 기능

그림 7에서 업계 채택을 위한 필수적인 감정적인 목표는 신뢰 할 수 있는 의도된 동작을 수행하는 사회 공학적 Qtum 시스템[34]에 대한 신뢰이다. 이 경우 신뢰는 목표를 달성하기 위해 기술을 사용하는 사람간 의 의존성과 관련된다. 우리는 경제 확산 가능성이 높고 광범위한 산업 확산을 위한 추가적인 감정적인 목표로 채택하기 쉽다고 생각한다. 전자는 Qtum 시스템을 사용하면 경제적인 투자 수익을 얻는 반면, 후자의 경우 Qtum을 사용하기 위한 진입 장벽이 낮아진다는 것을 의미한다.

Qtum 시스템의 모든 부분에 영향을 미치는 가치 제안과 관련된 품질 목표가 있다. 이러한 품질 목표는 상호 조직간 비지니스-프로세스 협업을 위한 참조 아키텍쳐[28]에서 나온다. 아래의 품질 목표는 [9, 17]에 따라 구성된다. 따라오는 품질 목표는 시스템 실행 시간 동안 식별 할 수 없다.

수정가능(Modifiable) 이란 Qtum 시스템 수명주기 동안 비즈니스 맥락으로 변경 및 적용됨을 의미한다. 또한 사용 소프트웨어를 정기적으로 업데이트를 하는 등 조직간 다른 기종 시스템 환경을 일치시킨다. ***통합가능***(Integrable) 시스템은 구성 요소 간의 인터페이스 프로토콜이 일치해야 하는 별도로 개발되고 통합된 구성 요소로 구성된다. 따라서 Qtum의 구성 요소간 의 통합을 보장해야 한다.

다음으로 런타임 중에 식별 가능한 Qtum의 품질 목표를 지정한다. ***상호운영성***(Interoperable) 이란 Qtum이 런타임에 계획, 물류, 생산, 외부 파트너 시스템 등과 같은 비즈니스 기능을 지원하는 시스템과 상호 운용되어야 한다는 것을 의미한다. 동적상호운영성 문제는 비즈니스, 개념 및 기술적 이질성이다. 보안(Secure)은 사용 및 서비스 거부에 대한 승인되지 않은 시도에 대한 저항과 신뢰 할 수 있는 사용자에게 훌륭한 평판을 제공한다. 보안, 신뢰 및 평판 문제를 해결하기 위해 Qtum에 대한 몇 가지 전략이 가능하다. 블록체인이 지원되는 인증 서비스는 협업 당사자를 확인하고, 네트워크 이벤트를 모니터 하며 검사하고 기록한다. 시스템의 통신은 암호화 될 수 있다. 고도로 자동화 된 협업을 위해서는 시스템이 전체 스마트 계약 수명주기를 포괄해야 한다.따라서

Qtum은 지루하고 반복적인 작업을 처리하는 동시에 인간이 창의적인 작업에 집중할 수 있도록 의미 있는 공동 작업 자동화의 가능성을 제공해야 한다. ***유연한***(Flexible) 협업은 다른 기종 데이터를 교환하는 다양한 파트너에 의한 활동을 규정하는 매우 역동적인 작업이다[25]. 따라서 Qtum은 다른 기종 개념과 기술을 조화시키는 다양한 상호 조직간 협업 시나리오를 가능하게 해야 한다. ***사용가능***(Usable)은 Qtum이 조직간 정보-물류 자동화에 사용하기 쉽고 세 개의 지역으로 분리되어야 함을 의미한다.오류 회피는 일반적으로 발생하는 공동 작업 오류를 예상하고 예방해야 한다. 오류 처리는 사용자가 오류를 복구 할 수 있도록 시스템을 지원한다. 학습능력은 사용자가 Qtum 시스템을 습득하는데 필요한 학습시간을 의미한다.

마지막으로 아키텍처별 품질 목표가 있다.***완전성***(Completeness)은 스마트 계약 수명주기 관리를 위한 일련의 구성 요소로 구성된 Qtum의 품질이다.***확장성***(Scalable) 이란 Qtum이 두개 이상의 공동 작업자를 하나의 구성으로 결합 할 수 있는 능력을 의미한다. ***적용가능성***(Applicable)은 Qtum이 상호-조직간 정보-물류 및 가치 이동을 자동화하는 수단이 된다는 것을 의미한다.이동성(Portable)은 Qtum이 비즈니스 도메인, 개념적 시스템 및 기술 시스템 인프라와 관련하여 산업 도메인 및 협업 이질성과 독립적인 정보-물류를 지원함을 의미한다.여기에는 모바일 장치도 포함됨을 유념해야 한다.***성능기준충족***(Performant)은 정보-물류 자동화를 위한 계산 및 통신 부담이 낮음을 의미한다.따라서 스마트 계약 수명주기의 모든 단계가 바람직한 응답 시간 내에서 그리고 컴퓨팅 능력에 대한 급격한 요구없이 수행되는 것이 중요하다.

3.2 수명 주기 관리 예제

3.1절의 목표 모델은 실행중인 해산물 식품 케이스에 투영 하기 위해 그림 8에 사상 된다.그림 8의 모델링 표기법은 비즈니스 프로세스 모델 및 BPMN 표기법[22] 이며 전체 라이프 사이클은 [18, 26, 27]에 공식화 되어 있다. 녹색원은 수명주기의 시작을 나타내고 빨간색원은 수명주기가 끝나는 것 나타낸다. +기호가 있는 직사각형은 3.1절의 수명주기 단계에서 해당하는 하위 프로세스이다. 하위 프로세스는 하위 레벨 비즈니스 프로세스 세부 사항을 숨기는 복합 활동이다.

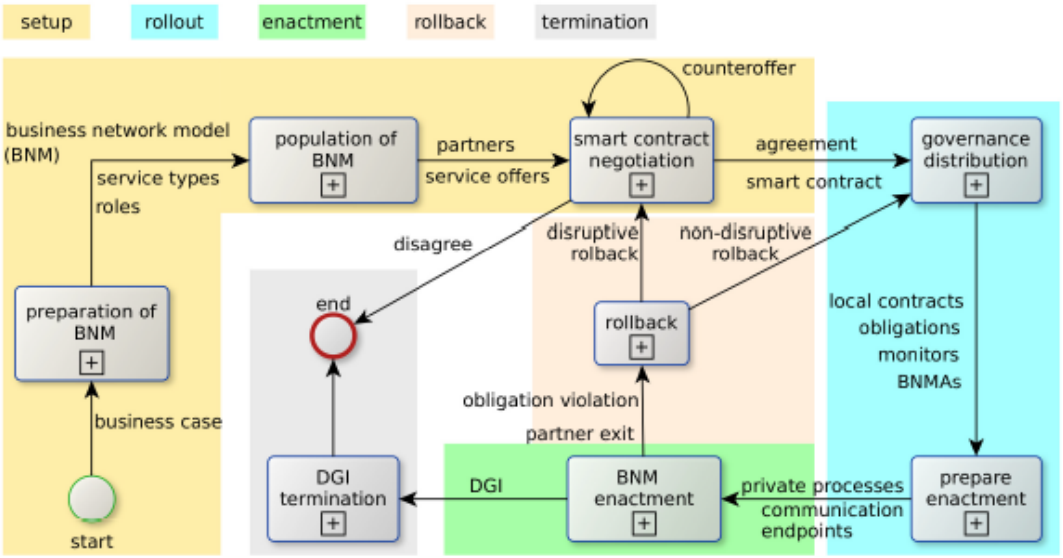

그림 8. Qtum 스마트-계약 수명주기 관리

그림 8의 각 스마트 계약 수명주기의 시작점은 상호-조직간 정보-물류 자동화가 필요한 해산물 운송의 비즈니스 사례이다.스마트 계약 체결을 위한 준비 플랫폼 역할을 하는 공동 작업 허브[29]가 있다고 가정하면 설계자는 서비스 유형이 역할과 함께 삽입되는 비즈니스 네트워크 모델(BNM) 용 템플릿을 작성한다.

BNM 템플릿이 생성 단계에 들어간다.각 서비스 유형과 관련된 역할을 스마트 계약(*bank2,seller,fridge2, buyer and bank1*)에서 협업하는 조직으로 채워진다. 여러 후보 조직이 특정 역할을 수행하기 위해 경쟁 할 수 있다는 점을 유의해야 한다. 역할을 충족 하려는 욕구를 강화하기 위해 잠재적인 파트너 조직은 역할이 속한 서비스 유형으로 서비스 제안을 일치시켜야 한다. 서비스 소비자는 제안서를 평가하고 서비스 제안이 수용 가능한지 여부를 결정할 수 있다.

모든 역할이 채워지고 서비스 유형이 수용 가능한 서비스 제공과 일치하면 스마트 계약 협상이 시작된다.우리는 실행중인 해산물 배달 사건의 당사자가 합의에 이르지 않고 설정 단계를 갑자기 끝내기를 원한다고 가정해보자. 대신 구매자는 해산물을 보관하는 컨테이너 내부의 온도와 관련된 의무를 소개하는 판매대를 제공한다.선적 컨테이너에는 Internet-of-Thins(IoT) [15] 센서가 장착되어 온도가 임계값을 초과하면 실시간으로 화주, 판매자 및 구매자에게 알려주는 것을 가정한다. 수정 제안은 해산물 식품의 품질 저하에 따라 가격인하를 규정한다.온도 변화로 해산물이 더 이상 소비되지 않을 경우, 구매자는 도착했을 때 구매를 거부 할 권리가 있다.수정 제안은 다른 모든 당사자가 수락하고 합의를 발생한다. 이는 계약 체결을 위한 전제 조건이다.스마트 계약은 분산 정부 기반 시설(Distributed governor infrastructure)를 추론해야 하는 조정 기관이다.따라서 운영중인 각 당사자는 각 계약의 의무가 추론되는 지역 계약서 사본을 받는다.예를 들어 운송인에 대한 의무는 해산물 운송 컨테이너의 내부 온도를 20도 이상 넘기지 않게 하는 것이다.의무사항

은 IoT 센서에 연결된 모니터 및 할당된 비즈니스 네트워크 모델 에이전트(Business-network model agents)가 관찰한다.

다음으로 모든 협동 당사자는 자신의 개별 프로세스를 신흥 기업(DGI)에 할당 할 수 있다.예를 들어 우리는 구매자가 먼저 Euro로 구매해야 하는 Bitcoin의 peer-to-peer 지불을 가정한다.*Bank1*을 통한 구매 및 지불에는 정부가 암호화화폐 사용에 대한 규제를 부과하는 준수 및 보고 단계로 구성된 프로세스가 포함된다.판매자의 *bank1*과 *bank2* 사이에 정보 교환을 가능하게 하려면 통신 종점을 설정해야 한다.그렇게 하면 판매자 규정 준수 데이터 관리가 자동화 된다.

Fridge1 의 도메인에서 온도 임계값 위반이 발생하면 할당된 BNMA가 이벤트를 확대 시키고 구매자는 위반 심각성을 확인한다.온도 위반이 일정 기간 지속될 경우 해산물 품질이 저하되어 구매자가 용인하는 저렴한 가격으로 성공적인 판매가 허용되면, 결과는 후자가 *fridge1*에 들어가는 다른 회사에 의해 더 냉동 시킬 것을 요구 할 수 있다.해산물이 심각하게 부패하여 대상 국가에서 판매 될 수 없다고 가정하면 구매자는 거래내역을 축소하는 파괴적인 롤백을 실시한다. 합의된 주에서 구매자와 함께 해산물 선적이 도착하고 *bank2*를 통해 판매자에게 지불이 완료되면 종료료 단계에서 DGI를 해산하고 모든 협력 업체를 방출한다.

다음으로 그림 8의 수명주기 관리가 조정하는 세부적인 공동 작업 요소간 의 관계를 설명한다.

4. 가치-전송 프로토콜

Qtum-프레임워크의 핵심 부분은 가치 제안에 따라 상호 조직간 정보-물류 및 가치 이동을 조정하는 가치-전송 프로토콜(value-transfer protocol, VTP)의 개념이다.이는 그림 7에 묘사되어 있다. 결과적으로 섹션 4.1에서는 VTP를 구성하는 프로세스 유형의 관계에 대해 설명한다. 4.2절에서는 유틸리티를 사용하여 VTP를 지정하는 특정 스마트 계약 언어의 필요성을 논의한다. 마지막으로 4.3절에서는 이더리움이 사용하는 VTP 지원 언어와 Solidity의 기능에 대해 설명한다.

4.1 상호-조직간 프로세스

VTP는 세 가지 유형의 공동 작업 프로세스로 구성된다. 그림 9는 3장에서 소개한 해산물 배달에 대한 BPMN 표기법의 간략화 된 BNM을 보여준다. BNM은 일련의 하위 프로세스가 조직의 역할을 나타내는 레이블과 함께 서비스 유형[12, 13]의 표시자라 가정한다.

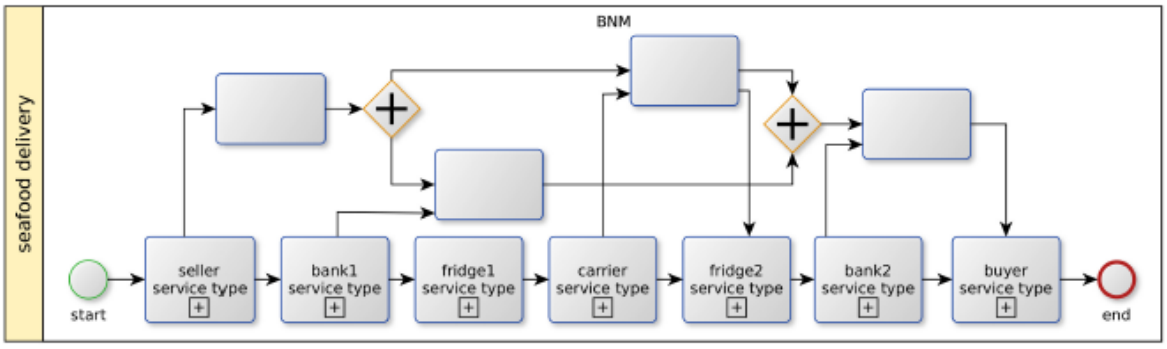

그림 9. Qtum BNM

BNM은 안무 제어 흐름을 수립하기 위해 서비스 유형 하위 프로세스를 연결하는 작업으로 구성된다고 가정한다.단순화를 위해 그림 9는 AND-분할 및 결합과 함께 레이블이 지정되지 않은 안무 작업을 묘사한다. BNM은 은행에 국제 통화 거래 준비를 알리는 해산물 판매자와 함께 시작하고 다음으로 운송업자가 목적지까지 운송하기 전에 해산물이 냉각된다.대상 국가에서는 해산물이 다시 냉각되고 지방 은행은 양국 간의 통화 거래를 처리한다. 마지막으로 구매자는 현지 판매를 위해 해산물을 받는다.

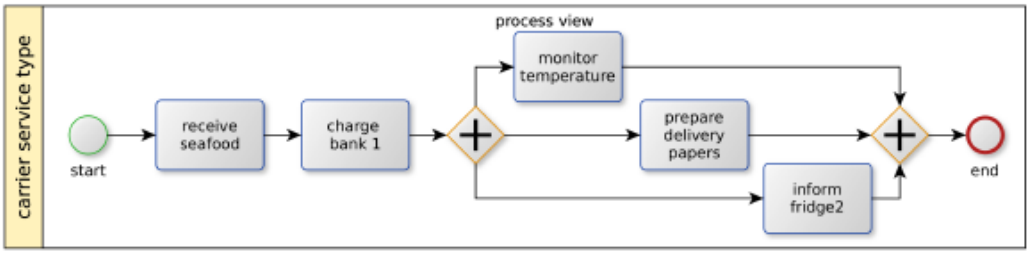

그림 10. 외부화 된 서비스 유형 프로세스 뷰

BNM의 운송 업체 하위 프로세스의 경우 해산물 운송 업체의 역할을 수행하기 위한 후보 조직이 여러 개 존재한다고 가정한다.그림 10은 서비스 유형 프로세스 보기의 형태로 하위 상세 검색을 단순화 한 예시를 보여준다[12, 13].그림 10의 단순화된 과정은 운송 업체가 원산지 국가의 냉장고에서 해산물을 받고 판매자의 은행에 청구하는 것으로 가정한다. 다음으로 세 개의 병렬 지점은 온도 모니터링, 전달 서류 준비 및 대상 회사의 냉각 회사에 동시에 알리는 것을 요구한다.단순화된 프로세스를 준수하겠다고 약속한 호부 조직만이 서비스 제공 업체가 될 수 있다.협업 허브[30]는 후자와 해당 서비스 제공 조직을 일치시키기 위해 서비스 유형 프로세스 뷰를 제공 할 수 있다.

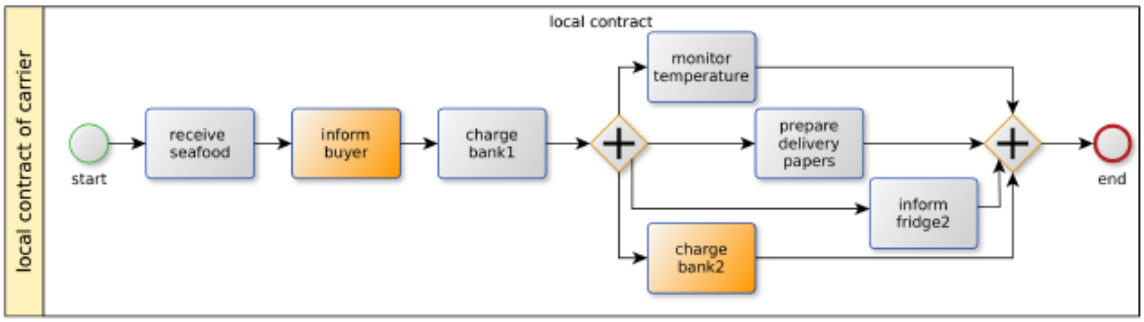

그림 11. 로컬 캐리어 계약

세번째 VTP요소인 그림 11은 캐리어가 내부적으로 사용하는 로컬 계약을 보여준다.그림 10의 서비스 유형 프로세스 뷰와 다르게 지역 계약은 구매자와 청구 은행2에 레이블을 표시하는 두개의 추가 작업으로 구성된다. 따라서 로컬 계약은 제정 행동과 관련한 서비스 유형 프로세스 뷰의 하위 클래스이다[12, 13]. 이는 프로세스 뷰의 모든 태스크는 외부적으로 경험되는 반면, 캐리어는 경쟁 우위를 구성하거나 외부 디스플레이에 관심이 없는 프라이버시 보장 방식으로 숨겨진 추가 단계를 삽입하는 옵션을 갖는다.

4.2 Qtum 스마트-계약 언어

4.1절의 VTP 시나리오를 지원하기 위해 현재 스마트 계약 언어인 Solidity는 포함된 개념 및 특성과 관련하여 필요한 유틸리티 수준을 갖지 않는다.대신 VTP관리를 위한 비교적 우수한 유틸리티를 가진 Qtum 스마트 계약 언어(Qtum smart contract language, QSCL)와 컴파일러를 개발하는 것이 목표이다.

그림 12는 High-level ASCL 개념과 속성을 설명한다. 4.1절의 VTP 시나리오는 전용 언어가 존재하는 eSourcing 프레임워크과 유사하다. 이는 시맨틱-웹 도메인에 대해 지정된 eSourcing Markup Language(eSML)[31] 이다.새로운 Qtum 가상 컴퓨터용 언어 컴파일러와 함께 QSCL을 만들기 위해 eSML의 개념과 속성을 블록체인 도메인에 투영하려고 한다.

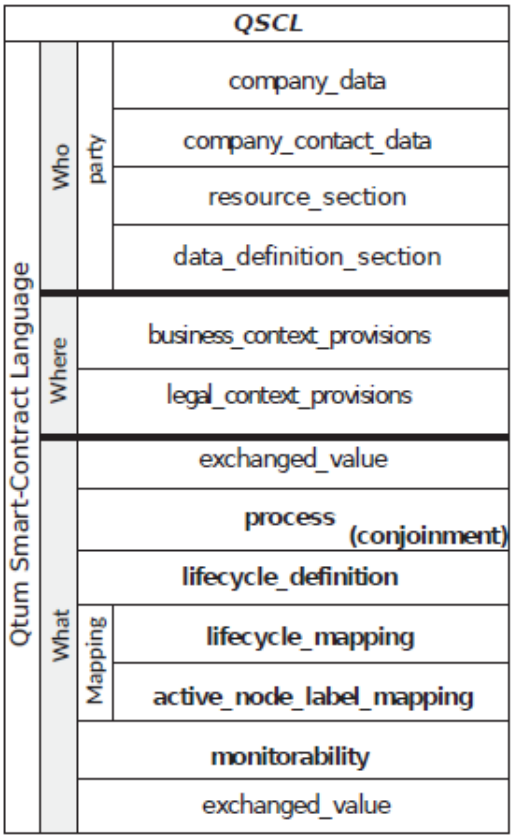

그림 12. 미래의 Qtum 스마트 계약 언어의 속성 및 개념

간단히 말하면 독자에게 더 자세한 내용을 알리는 동안 그림 12의 속성은 개념적인 질의형식에 따라 구성된다.하나의 QSCL-인스턴스는 BNM정의와 유사하다(그림 9). QSCL의 `Who`개념은 관련된 자원 및 데이터 정의와 함께 계약 당사자를 고유하게 정의하기위한 구성을 포함한다.`Where`개념은 비즈니스 계약을 지정하고 특정 스마트 계약이 보유하는 법적 문맥 조항도 지정한다.`What`개념은 교환된 값과 서비스 유형 프로세스 뷰(그림 10)를 이러한 프로세스 뷰와 기본 태스크에 대한 수명주기 정의와 함께 정의 할 수 있게 한다.따라서 QSCL 인스턴스의 어떤 부분에서 몇 가지 서비스 유형 프로세스 뷰를 그림 9와 비교하여 정의 할 수 있다.마지막으로 결합 생성자는 상소-조직간 데이터 흐름을 위해 특별히 정의 된 교환 채널이다.***모니터링 가능성***(Monitorability) 구성은 폴링 또는 메시징 원칙을 사용하는 전용 작업 모니터링의 유연한 정의를 허용한다.

4.3 비교 토론

스마트 계약 존재론[31]을 사용하여 우리는 비공식적으로 Qtum-프레임워크에 대해 구축한 기존의 Solidity 대 QSCL의 적합성을 조사한다. 일반적인 관찰로 Solidity 는 JavaScript 와 유사한 구문을 사용하여 대부분 low-level 블록체인 조작 명령에 중점을 둔 언어이다. 여전히 third-party

API 를 가져와서 외부 함수 호출을 수행 할수 있다.소위 Solidity의 외부 함수는 다른 계약 및 거래내역을 통해 호출 할 수 있는 스마트 계약 인터페이스의 일부이다.

Turin-completeness 때문에 QSCL이 구현하는 스마트 계약 존재론의 모든 개념과 특정에 대한 성가신 지원을 정의하는 것이 원칙적으로 가능하다.그러나 패턴 기반 설계, 프로세스 인식, 프로세스 일치 등과 같은 개념은 Solidity 에서 어떤 방식으로 든 채택되지 않는다.번거로운 해결 방법을 개발하는 것과 관련하여, 최근의 학회 논문[43]은 Solidity를 사용하여 신뢰할 수 없는 비즈니스 프로세스 모니터링 및 스마트 계약의 실행 가능성을 입증한다.

Solidity는 QSCL의 설계 초기[31]와는 달리 형식적 검증 수단으로 역사적으로 뒷받침되지 않았다고 강조해야 한다.공식적으로 검증 가능한 표현이 없으면 계약서가 정확하고 보안 문제가 없는 경우 제정안을 미리 알 수 없다. Solidity 관련 보안 사건은 Why, Solidifier 또는 Casper와 같은 검증 도구의 개발 및 적용을 최근에 시작하여 이더리움에 대한 작업증명에서 지분증명으로 전환할 가능성이 높다.

5. 결론

이 백서는 새로운 스마트 계약 블록체인-기술 솔루션을 위한 Qtum-프레임워크를 제공한다.우리는 지분증명 유효성 검사를 사용하는 특정 Qtum 거래내역-처리 구현을 설명한다.또한 Qtum은 Bitcoin 미사용 거래내역 출력 프로토콜과 함께 이더리움 가상 컴퓨터(EVM)를 통합한다. Qtum EVM은 항상 이전 버전과의 호환성을 유지함을 유의해야한다.또한 Qtum-프레임워크는 스마트 계약 수명주기 관리가 공동 작업자의 적절한 보안 검사를 지원한다는데 중요하다는 것을 인식한다. Qtum 수명주기 관리를 지원하기 위해 현재 공용어인 Solidity는 적합성이 부족하다.결과적으로 새로운 Qtum-프레임워크에는 향상된 유틸리티가 포함된 새로운 스마트 계약 언어가 필요하다.

Qtum에 지분증명을 채택하면 여전히 작업증명을 사용하는 이더리움에 비해 계산 작업이 크게 절약된다.이더리움은 지분증명 방식을 채택할 계획이지만 새로운 버전이 공개 될지는 불분명하다.또한 미사용 거래내역 출력의 사용은 이더리움 계정관리와 비교하여 더 확장이 가능하다.간단한 지불 검증과 함께 Qtum은 이미 스마트 계약 모바일 장치 솔루션을 개발하고 있다.이더리움 솔루션을 확장하지 않으면 모바일 솔루션을 사용 할 수 없지만 Qtum은 모바일 전략을 통해 민주화되고 고도로 분산된 지분증명 거래내역 검증을 달성하는 것을 목표로 하고 있다.

Qtum-프레임워크는 미래의 개발이 충족해야 하는 품질 기준을 명확하게 이해하고 있다.기능 요구 상황과 관련하여 Qtum은 스마트 계약 수명주기 관리를 위한 응용프로그램 계층을 개발할 계획이다.가장 중요한 것은 그런 수명주기 관리가 협력 당사자를 대상으로 최근 이더리움이 경험한 것과 같은 보안 침해를 줄이기 위해 중요하다는 점이다.

Qtum의 정보-물류를 위한 가치-전송 프로토콜은 여러 협력 조직을 구성하는 비즈니스 네트워크 모델로 구성된다.이는 비즈니스 네트워크 모델에서 서비스 유형 프로세스 뷰의 지정된 런타임 동

작과 일치해야 하는 로컬 계약이 있는 서비스를 제공 할 수 있다.다층의 스마트 계약 관리 레이어를 사용하면 협력 업체가 로컬 계약의 확장 단계를 숨김으로써 경쟁 우위를 차지하는 비즈니스 비밀의 개인 정보를 보호 할 수 있다.

요약하면, Qtum-프레임워크는 스마트 계약이 광범위한 사용자 채택을 달성하기 위한 필수 품질 요구 사항을 고려해야 하는 사회 공학 산출물이라는 것을 인식된다. Qtum 응용프로그램을 사용하여 진행중인 실제 산업 프로젝트는 지속적으로 경험적 요구 사항을 습득한다.고도로 분산된 지분증명 거래내역 처리를 지원하는 모바일 전략은 최첨단 기술분야에서 상당한 발전을 목표로 한다.아직은 Qtum이 스마트 계약 수명주기 관리를 위해서는 현재 솔루션이 충분히 주의를 기울이지 않는 정교한 프론트 엔드 사용자 경험을 갖춘 어플리케이션 계층 개발이 필요하다는 사실을 인정한다.

참고문헌

1. A.M Antonopoulos. Mastering bitcoins, 2014.
2. I. Bentov, A. Gabizon, and A. Mizrahi.Cryptocurrencies Without Proof of Work, pages 142?157. Springer Berlin Heidelberg, Berlin, Heidelberg, 2016.
3. K. Bhargavan, A. Delignat-Lavaud, C. Fournet, A. Gollamudi, G. Gonthier, N. Kobeissi, N. Kulatova, A. Rastogi, T. Sibut-Pinote, N. Swamy, and S. Zanella-B´eguelin. Formal veri?cation of smart contracts: Short paper. InProceedings of the 2016 ACM Workshop on Programming Languages and Analysis for Security, PLAS '16, pages 91?96, New York, NY, USA, 2016. ACM.
4. A. Biryukov and D. Khovratovich. Equihash: Asymmetric proof-of-work based on the generalized birthday problem.Proceedings of NDSS?aA˘Z´16, 21?24 February 2016, San Diego, CA, USA. ISBN 1-891562-41-X, 2016.
5. B. Bisping, P.D. Brodmann, T. Jungnickel, C. Rickmann, H. Seidler, A. Stu¨ber, A. Wilhelm-Weidner, K. Peters, and U. Nestmann. Mechanical veri?cation of a constructive proof for ?p. InInternational Conference on Interactive Theorem Proving, pages 107?122. Springer, 2016.
6. O. Bussmann.The Future of Finance: FinTech, Tech Disruption, and Orchestrating Innovation, pages 473?486. Springer International Publishing, Cham, 2017.
7. C. Cachin. Architecture of the hyperledger blockchain fabric. InWorkshop on Distributed Cryptocurrencies and Consensus Ledgers, 2016.
8. K. Christidis and M. Devetsikiotis. Blockchains and smart contracts for the internet

of things.IEEE Access, 4:2292?2303, 2016.
9. L. Chung, B.A. Nixon, E. Yu, and J. Mylopoulos.Non-functional requirements in software engineering, volume 5. Springer Science & Business Media, 2012.
10. K. Croman, C. Decker, I. Eyal, A.E. Gencer, A. Juels, A. Kosba, A. Miller, P. Saxena, E. Shi, E. Gu¨n Sirer, D. Song, and R. Wattenhofer.On Scaling Decentralized Blockchains, pages 106?125. Springer Berlin Heidelberg, Berlin, Heidelberg, 2016.
11. N. Emmadi and H. Narumanchi. Reinforcing immutability of permissioned blockchains with keyless signatures' infrastructure. InProceedings of the 18th In-
26 Alex Norta
ternational Conference on Distributed Computing and Networking, ICDCN '17, pages 46:1?46:6, New York, NY, USA, 2017. ACM.
12. R. Eshuis, A. Norta, O. Kopp, and E. Pitkanen. Service outsourcing with process views.IEEE Transactions on Services Computing, 99(PrePrints):1, 2013.
13. R. Eshuis, A. Norta, and R. Roulaux. Evolving process views.Information and Software Technology, 80:20 ? 35, 2016.
14. D. Frey, M.X. Makkes, P.L. Roman, F. Ta¨ıani, and S. Voulgaris. Bringing secure bitcoin transactions to your smartphone. InProceedings of the 15th International Workshop on Adaptive and Re?ective Middleware, ARM 2016, pages 3:1?3:6, New York, NY, USA, 2016. ACM.
15. J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami. Internet of things (iot): A vision, architectural elements, and future directions.Future Generation Computer Systems, 29(7):1645 ? 1660, 2013.
16. A. Kiayias, I. Konstantinou, A. Russell, B. David, and R. Oliynykov. A provably secure proof-of-stake blockchain protocol, 2016.
17. G. Kotonya and I. Sommerville.Requirements engineering: processes and techniques. Wiley Publishing, 1998.
18. L. Kutvonen, A. Norta, and S. Ruohomaa. Inter-enterprise business transaction management in open service ecosystems. InEnterprise Distributed Object Computing Conference (EDOC), 2012 IEEE 16th International, pages 31?40. IEEE, 2012.
19. L. Luu, D.H. Chu, H. Olickel, P. Saxena, and A. Hobor. Making Smart Contracts Smarter. InProceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, CCS '16, pages 254?269, 2016.
20. L. Luu, V. Narayanan, C. Zheng, K. Baweja, S. Gilbert, and P. Saxena. A secure sharding protocol for open blockchains. InProceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, CCS '16, pages 17?30, New York, NY, USA, 2016. ACM.
21. J. Marshall. Agent-based modelling of emotional goals in digital media design projects.International Journal of People-Oriented Programming (IJPOP),

3(1):44?59, 2014.
22. Business Process Model. Notation (bpmn) version 2.0.Object Management Group speci?cation, 2011.http://www.bpmn.org.
23. S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system.Consulted, 1(2012):28, 2008.
24. N.C. Narendra, A. Norta, M. Mahunnah, L. Ma, and F.M. Maggi. Sound con?ict management and resolution for virtual-enterprise collaborations.Service Oriented Computing and Applications, 10(3):233?251, 2016.
25. A. Norta.Exploring Dynamic Inter-Organizational Business Process Collaboration. PhD thesis, Technology University Eindhoven, Department of Information Systems, 2007.
26. A. Norta.Creation of Smart-Contracting Collaborations for Decentralized Autonomous Organizations, pages 3?17. Springer International Publishing, Cham, 2015.
27. A. Norta.Establishing Distributed Governance Infrastructures for Enacting Cross-Organization Collaborations, pages 24?35. Springer Berlin Heidelberg, Berlin, Heidelberg, 2016.
28. A. Norta, P. Grefen, and N.C Narendra. A reference architecture for managing dynamic inter-organizational business processes.Data & Knowledge Engineering, 91(0):52 ? 89, 2014.
Smart-Contract Information- & Value Logistics 27
29. A. Norta and L. Kutvonen. A cloud hub for brokering business processes as a service: A "rendezvous" platform that supports semi-automated background checked partner discovery for cross-enterprise collaboration. InSRII Global Conference (SRII), 2012 Annual, pages 293?302, July 2012.
30. A. Norta and L. Kutvonen. A cloud hub for brokering business processes as a service: A "rendezvous" platform that supports semi-automated background checked partner discovery for cross-enterprise collaboration.Annual SRII Global Conference, 0:293?302, 2012.
31. A. Norta, L. Ma, Y. Duan, A. Rull, M. Ko?lvart, and K. Taveter. eContractual choreography-language properties towards cross-organizational business collaboration.Journal of Internet Services and Applications, 6(1):1?23, 2015.
32. A. Norta, A. B. Othman, and K. Taveter. Con?ict-resolution lifecycles for governed decentralized autonomous organization collaboration. InProceedings of the 2015 2Nd International Conference on Electronic Governance and Open Society: Challenges in Eurasia, EGOSE '15, pages 244?257, New York, NY, USA, 2015. ACM.
33. Aafaf Ouaddah, Anas Abou Elkalam, and Abdellah Ait Ouahman.Towards a Novel Privacy-Preserving Access Control Model Based on Blockchain Technology

in IoT, pages 523?533. Springer International Publishing, Cham, 2017.
34. E. Paja, A.K. Chopra, and P. Giorgini. Trust-based speci?cation of sociotechnical systems.Data & Knowledge Engineering, 87:339 ? 353, 2013.
35. J. Poon and T. Dryja. The bitcoin lightning network: Scalable o?-chain instant payments, 2015.
36. M. Rosenfeld. Overview of colored coins.White paper, bitcoil. co. il, 2012.
37. P. Serguei. A probabilistic analysis of the nxt forging algorithm.Ledger, 1:69?83, 2016.
38. L. Sterling and K. Taveter.The art of agent-oriented modeling. MIT Press, 2009.
39. T. Tenso, A. Norta, and I. Vorontsova. Evaluating a novel agile requirements engineering method: A case study. InProceedings of the 11th International Conference on Evaluation of Novel Software Approaches to Software Engineering - Volume 1: ENASE,, pages 156?163, 2016.
40. P Vasin. Blackcoina?A˘ Z´s proof-of-stake protocol v2, 2014.
41. M. Vukoli´c. The quest for scalable blockchain fabric: Proof-of-work vs. bft replication. InInternational Workshop on Open Problems in Network Security, pages 112?125. Springer, 2015.
42. M. Vukoli´c.The Quest for Scalable Blockchain Fabric: Proof-of-Work vs. BFT Replication, pages 112?125. Springer International Publishing, Cham, 2016.
43. I. Weber, X. Xu, R. Riveret, G. Governatori, A. Ponomarev, and J. Mendling. Untrusted Business Process Monitoring and Execution Using Blockchain, pages 329?347. Springer International Publishing, Cham, 2016.
44. G. Wood. Ethereum: A secure decentralised generalised transaction ledger. Ethereum Project Yellow Paper, 2014.