kadena

scalable blockchain | smarter contracts

Kadena 퍼블릭 블록체인

프로젝트 요약 백서

Version 1.2 November, 2018

Will Martino
will@kadena.io

Stuart Popejoy
stuart@kadena.io

서론

본 백서는 Kadena 퍼블릭 블록체인 기술의 기본적인 개요로 스마트 컨트랙트 언어 Pact와 새로운 병렬 체인 작업 증명(PoW) 구조인 Chainweb에 대해서 설명한다. 두 기술은 결합되어 높은 트랜잭션 처리량과 하드포크 없는 스마트 컨트랙트의 업그레이드를 가능케 하며 이로써 새로운 시대에 필요한 비즈니스 워크플로우 개선과 거버넌스 매커니즘(Governance Mechanism)을 확실하게 제공한다.

목차

Pact를 이용한 더 안전한 스마트 컨트랙트

Pact는 블록체인에서 가치가 높은 비즈니스 워크플로우를 안전하게 수행하기 위한 솔루션으로 설계되었다. 이더리움의 솔리디티(Solidity)와 같은 언어는 실패 시 명확한 오류 메시지를 포함하는 비즈니스 규칙 적용, 데이터베이스 스키마의 모델링 및 유지 관리, 사용자의 신중한 작업 수행 승인 등의 비즈니스 애플리케이션의 일일 작업이 되는 주요 기능이 부족하다. 이러한 프로세스에는 전문성이 필요하다. 디자인이나 중요 기능 적용 등을 각 개발자의 재량에 맡기게 되면 생산성 저하는 물론이고 무엇보다도 버그와 취약점 공격을 유발한다. Pact는 이러한 필수 기능들을 언어 내에 통합해 코드 작성과 테스트를 더욱 수월하게 하며 코드에 대한 이해도를 높인다.

Pact는 스마트 컨트랙트가 *사람이 읽을 수 있고 컴퓨터에 의한 검증*이 가능해야 한다는 굳은 신념을 가지고 설계되었다. 우리는 고급 사용자조차도 이해할 수 없는 로우 레벨 바이트 코드의 대규모 스트림의 저장과 호출이 필요한 EVM과 같은 바이트 코드 기반 인터프리터 사용에 강한 이의가 있다. 대신에 Pact는 인터프리티드 언어로서 개발자가 작성한 코드 그대로 블록체인에 저장되며 언제나 쓰인 그대로 읽힐 수 있다. 이로써 Pact의 접근성은 개발자와 비개발자 모두에게 높아지며 기술 방면에 정통한 법률가와 기업 임원까지 스마트 컨트랙트 코드의 비즈니스 로직을 쉽게 검토할 수 있게 한다.

Pact가 안전에 주안점을 두는 것은 비트코인 스크립트의 영향이 짙은데 이는 해당 스크립트가 최소한의 기능 집합으로 최대한의 확실성을 코인 전송에 더해지게끔 디자인되었기 때문이다. 이를 염두에 두어 Pact는 의도적으로 튜링 불완전(Turing-incomplete) - 재귀 또는 무한 루핑이 불가능하다.[1] 하지만 Pact에서 루프 종결형 리스트 함수 컨셉들인 map, filter, fold는 충분히 가능하다. 우리는 블록체인에서 사용될 대다수의 작업들은 튜링 완전(Turing-complete) 기능들이 불필요하다고 믿으며 이 기능이 절실히 필요한 작업(경로 찾기 또는 복합적인 가격 책정 등)은 자원에 제약이 있는 블록체인 환경에서 사용하기 부적합하다고 판단한다. 이러한 작업들은 예측 할 수 없는 양의 컴퓨팅 파워를 필요로 하며 이로 인한 예를 들자면 이더리움에서는 언제든 "가스가 모자라"[2] 작업이 실패 할 수 있다.

컨트랙트 코드의 정형 검증

이더리움 생태계에서 떠오른 취약점과 공격들로 인해 스마트 컨트랙트를 통한 중요 비즈니스 프로세스의 자동화의 위험성은 명백해졌다. *정형 검증(Formal Verification)*은 스마트 컨트랙트의 안전성과 확실성을 대폭 증가시킬 강력한 솔루션이다. 정형 검증을 통하면 코드는 해당 기능의 수학적 모델으로 변환되며 해당 모델의 속성이 어떠한 조건을 충족하는지를 증명한다. 이 기술은 무한대의 입력 공간과 입력 상태에 걸쳐 동작이 올바른지에 대한 유효성을 검증하며 통상적인 소프트웨어 테스트와는 근본적으로 다르다. 표준 소프트웨어 개발 실무 관행(유닛 테스트 등)으로는 알려진 상황의 테스트만 가능할 뿐이다.

정형 검증 및 SMT(Satisfiability Modulo Theories: 충족 가능 모듈로 이론)는 일반적인 프로그래머의 전문 기술을 웃도는 수준이 요구되는 고도로 전문화된 컴퓨터 과학 연구 분야이다. Pact의 검증 시스템은 프로그램이 증명성 있게 종결되고 타입체킹됨으로서 먼저 구축되며 이후 SMT-LIB2 언어로 컴파일링을 함으로서 Z3 정리 증명에 사용될 수 있다.

Pact는 그 후 스마트 컨트랙트에 임베드될 강력하고 명료한 "mini-language" 또한 제공한다. 이것은 "@model" 태그로 구분이 되며 Z3 환경이 위반을 시도할 *속성*을 표현한다. 예를 들어 데빗과 크레딧이 '0'과 일치하도록 강요하는 선언으로 이중 지불을 방지하는 잔액 송금 시스템을 표현할 수 있다. 이 DSL은 읽기 쉬우며 기술에 문외한인 관계자도 검증 규칙의 완전함을 확인할 수 있게 해준다.[3]

---

[1] 이더리움(Ethereum) DAO 해킹은 재귀 기반의 취약점 공격 측면이 특히 강하다.
[2] "가스(gas)"는 이더리움에서 계산 및 데이터 사용을 조절하기 위한 암호화폐 결제를 말한다. Pact 컨트랙트도 Kadena 퍼블릭 블록체인의 일부 가스 모델에 의해 조정을 받는다.
[3] 간단하면서도 이해하기 쉬운 코드와 증명 식에 정형 검증을 접목할 수 있는 시스템은 오늘날에도 제공되거나 제안된 바 없다. 그에 비해서 Tezos를 사용하려면 개발자는 정식으로 지정된 로우 레벨 바이트 코드로 스마트 컨트랙트을 작성하고 Coq 정리 증명기를 사용해서 증명을 완성해야 한다.

거버넌스가 내장된 스마트 컨트랙트

일반적으로 블록체인은 프로토콜 레벨 업데이트 제공을 위해 업그레이드가 반드시 필요한 *프로토콜 클라이언트 소프트웨어*에서 실행된다. 이를 달성하기 위해 블록체인은 *하드포크(hard fork)* 과정을 거쳐 네트워크의 모든 참여자들이 새 버전의 시스템으로 강제 전환하도록 한다. 하드포크는 블록체인 원장 및 프로토콜을 어떤 면에서든 변경할 수 있으며 이때 유일한 제약 사항은 해당 커뮤니티 멤버들의 비공식적인 합의이다.

이더리움에서의 스마트 컨트랙트는 시스템에 로드됨에따라 주소(전송자의 퍼블릭 키 기반으로 생성)로 레퍼런스되며 한번 설치된 이상 절대로 업그레이드가 불가능하다.[4](한 시점에서는 이더리움이 이 사실을 "코드는 곧 법"이라는 문구로 대중화했다) 하지만 현실에서 스마트 컨트랙트가 업그레이드 되지 않는다는 점은 블록체인에 나타난 치명적인 취약점을 고치려 원장 항목을 덮어쓰는 심각한 프로토콜 레벨 하드포크 남용으로 이어졌다. 프로토콜에 국한된 하드포크는 논란거리가 되는 한편(비트코인 Segwit과 블록 사이즈 이슈와 같은), *원장 항목(컨트랙트 코드가 저장되는) 수정*을 위한 하드포크는 원장 데이터에 관한 중앙 집중식 권한을 표명하며 이는 블록체인의 신뢰없는, 또 탈중앙화된 철학에 상반된다.[5]

단언컨대 성숙한 스마트 컨트랙트 시스템이라면 하드포크를 하지 않고도 컨트랙트의 업그레이드를 반드시 지원해야 한다. 중요한 문제를 해결하고 전략적 향상을 꾀하기 위해서라면 컨트랙트의 수명 주기상 어느 지점에서든 업그레이드가 가능하여야 한다. Pact의 스마트 컨트랙트는 현재 키셋(퍼블릭 키 서명자에게 작업 권한을 부여하는 규칙)을 통해 거버넌스 지정을 요구하며 곧 범용적인 거버넌스 기능으로 대체된다. 키셋은 단일 – 다중 키 서명 작업을 제공하고 범용적인 거버넌스 기능은 이를 이해관계자 투표와 같은 완전히 탈중앙화된 모델으로 확장할 것이다. Pact 컨트랙트의 첫 작성자는 자신이 원하는 거버넌스 구조를 모델링할 자유를 얻게 된다.[6]

오라클과 서비스

로버스트한 블록체인 서비스 환경에서 스마트 컨트랙트는 신뢰할 수 있는 외부 소스인 *오라클*에서 데이터를 가져올 방법이 필요하다. 주가 정보 입수 또는 높은 강도의 계산 실행과 같은 오라클 프로세스는 체인 외부에서 실행된다. 이러한 프로세스는 퍼블릭 키 시그니처와 같은 출처 증명(Proof of Provenance)으로 인증된 데이터를 반환한다. *푸시 기반(push-based)* 접근법에서 오라클 소스는 컨트랙트에서 쿼리되는 주기적인 데이터를 발행한다. *풀*

[4] 스마트 컨트랙트 코드 호출에 간접 참조를 도입하는 소프트웨어 기술은 이 문제에 호재로 작용할 수 있지만, 복잡도가 증가하므로 여전히 근본적인 문제의 "우회적 해결"에만 도움이 된다.
[5] 버그 바운티 포상제가 도움이 되고 해결의 실마리를 제공하는 "파일럿 단계"를 사용하는 Bancor의 사례처럼 이 문제를 대하는 현재의 접근법은 임시변통에 불과하며, 파일럿 단계 후에 컨트랙트은 업그레이드가 영구적으로 불가능해지므로 중요한 문제를 해결하려면 하드포크가 불가피하다.
[6] Pact 거버넌스 기능은 데이터베이스에서 발생 및 기록되어온 혹시 모를 일부 표결 프로세스의 유효성을 검증함으로써 업그레이드 시도의 통과-실패로 작용한다.

기반(pull-based) 접근법에서는 스마트 컨트랙트가 외부 정보를 요청함으로서 오라클 프로세스를 개시한다.

기존 스마트 컨트랙트 언어는 풀 기반 오라클 프로세스를 구현하는 데에 사용자 지정 코드가 필요하다 – 이는 평균 개발자 수준을 넘어서는 복잡한 프로그래밍 작업이다. 이러한 상호 작용들은 데이터베이스에서 미해결된 요청 트래킹, 무응답 처리, 에스크로 결제 관리, 프로세스 완료 후 정리 등이 요구된다.

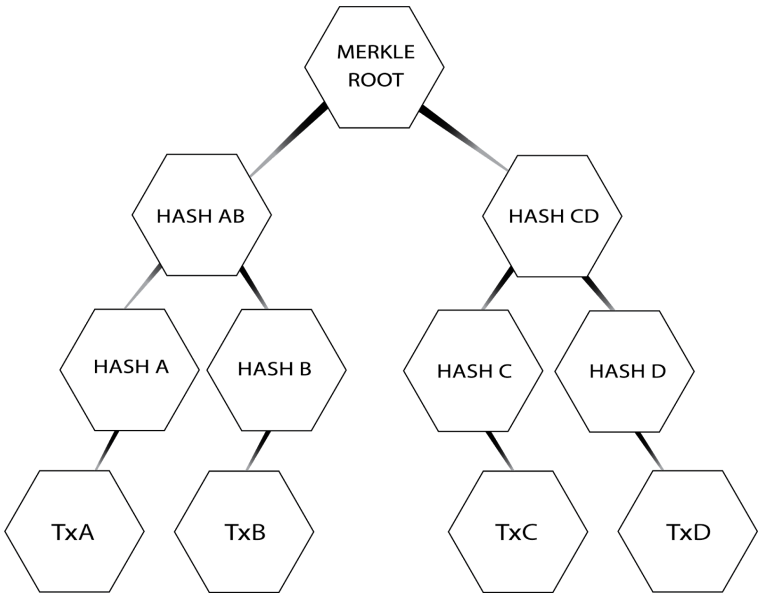
Pact는 이러한 오라클 프로세스를 pacts라는 기능으로 쉽고 안전하게 자동화한다. Pacts는 스텝별로 일시 중단(yield) 및 계속하기(resume)를 가능케하는 함수로서 일종의 다중 단계 트랜잭션이다. 스텝 실패시 롤백이 지정되어 변경 내용을 실패 후에 되돌린다.

오라클 지원 이외에도 Pact는 "전방향"으로 접근 할 수 있는 서비스를 디자인했다. Pact 스마트 컨트랙트의 함수는 자동으로 프론트엔드 REST API의 엔드포인트로 작동하며 Pact의 데이터는 기본 JSON 형식으로 표시된다. 이름 기반으로 저장되어 다른 스마트 컨트랙트들에게도 “horizontal"한 서비스 API로 작동하며, 블록체인 트랜잭션 안에서 서로 간에 함수 호출을 쉽게 만든다. 마지막으로 모든 데이터는 관계형 데이터베이스 백엔드에 직접 기록될 수 있으므로 다운스트림 시스템과의 통합과 심층 분석을 위한 데이터 내보내기를 수월하게 한다.

Chainweb

Chainweb은 꽈배기(braided) 체인으로 구성되어있으며 동일한 화폐를 채굴하며 유동성을 교환하는 새로운 *병렬 체인* 작업 증명(PoW) 구조이다. 기존의 작업 증명 구조들과 달리 Chainweb은 막대한 처리량을 제공한다. 1,250개의 체인부터 시작하여 초당 10,000개 이상의 트랜잭션을 실행하는 한편 작업 증명의 사기와 검열에 대한 강력한 회복력 또한 유지한다.

Chainweb의 디자인은 비트코인의 SPV(간편 결제 검증) 기능을 Kadena의 스마트 컨트랙트에 추가하려는 아이디어에서 시작되었다. SPV는 사용자가 전체 블록체인을 프로세스하지 않고 특정 트랜잭션만 검증할 수 있게 하는 라이트 클라이언트(light client) 지원으로도 알려져있다. 이는 블록체인에서 머클 증명을 쿼리함으로써 가능하다. 이러한 증명에서는 트랜잭션 해시와 기록들이 트리 자료구조의 리프(leaf)와 가지(branch) 노드로 저장된다. 맨 밑의 리프 노드로 시작해서 이웃하는 트랜잭션 해시들은 결합되어 트리의 새로운 노드가 되고 가장 최근의 트랜잭션 증명은 루트(root)가 된다.



Kadena의 퍼블릭 블록체인에서는 스마트 컨트랙트 SPV가 멀티스텝 pacts를 이용하여 *가상통화 간 교환을 자동화시킨다*. 예를 들어 앨리스가 자신의 Kadena 코인을 밥의 비트코인과 교환하고자 한다. 앨리스는 Kadena 코인을 에스크로하는 것으로 pact를 시작하고, 밥은 앨리스의 비트코인 주소로 전송을 했다는 머클 증명으로 응답한다. 해당 pact는 트랜잭션 기록의 증명을 검증하고 밥의 Kadena 계좌로 에스크로 자금을 양도한다.

이러한 교환을 가능하게 하려면 Pact는 비트코인, 이더와 같은 외부 가상통화의 머클 증명을 직접적으로 지원해야한다. 같은 Kadena 간의 pacts에서는 두 체인이 병렬으로 실행될 수 있으며 각 체인은 각자의 코인을 채굴하면서 SPV를 통한 신뢰없는 교환이 가능해진다.[7] 두개의 병렬체인으로 전체 트랜잭션 처리량은 두배가 된다.

Chainweb 보안

각기 다른 가상화폐간의 교환을 위해서는 머클 증명의 확인이 필요하며 이를 위해 기존 루트와 외부 가상화폐의 루트를 “연결(link)”해야 한다. 스마트 컨트랙트는 인터넷 접근 권한이 없으므로 신뢰받는 “oracle”이 머클 루트를 제공해야 한다. 하지만 같은 Kadena 간의 교환시에는 반대 체인의 이전 머클 루트를 블록 헤더에 게시할 수 있으므로 한 체인을 기타 체인의 머클 루트 "oracle"으로 지정하는 효과적인 체인 합의(chain consensus)가 가능하다.

여기서 각 체인 루트의 단순 트랜잭션 증명만으로는 부족하다. 전 세계에서 보존되는 코인들은 해당 체인들의 트랜잭션 기록들이 담긴 단일 뷰가 필요하다. 이를 달성하기 위해서 각 체인은 상대 체인의 머클 루트를 자체 루트로 해시한 후 해당 루트를 검사하여 가지가 갈라져 나오지 않는다는 것을 확인한다. 이때 체인들의 “꽈배기”는 유닛으로만 공격당할 수 있다. 악의적으로

[7] 복제 계좌를 막기 위해 양도자는 인출된 체인의 코인을 파괴한 후 입금된 체인에 해당 코인을 만들어 "광역"으로 코인을 보존한다.

어느 한 체인을 포크하기 위해서 공격자는 정직한 채굴자보다 빠르게 *양쪽 체인* 모두 해시해야 한다.

우리는 two-체인 디자인을 *n*-체인으로 확장함으로써 Chainweb이란 해결책에 도달했다. "꽈배기" 체인은 인접 체인의 머클 증명을 그래프 레이아웃으로 통합하여 증명이 시스템의 기타 모든 체인에 빠르게 전파되도록 한다.(블록 깊이의 최대 한도 내) 차수-지름 문제(degree-diameter problem)의 솔루션에 따르면 1,250 체인이 단 3개 블록만에 광역 전파를 달성하는 것으로 알려졌다.[8] 확인된 깊이(confirmation depth)를 늘리면 더 큰 구성이 가능해진다. 각 추가 체인을 통해서 처리량은 선형으로 증가하며 Chainweb "꽈배기" 체인의 확인된 깊이(confirmation depth)까지의 악의적인 포크를 견뎌내는데에 필요한 해시 속도는 실행 중인 모든 체인의 누적 해시 속도에 수렴한다.

중요한 점은 체인이 파일 저장소같은 특정한 작업에 *특화될* 수 있다는 것이다. 개발자는 스마트 컨트랙트이 실행될 체인을 지정함으로써 해당 작업의 처리량 요구 사항을 제공할 수 있다.

통합하여, 이러한 혁신은 예상치 못하는 수요 폭증의 버팀목이 되고, 동일한 회복력으로 작업을 더 효율적으로 수행하며 완전히 새로운 종류의 비즈니스 애플리케이션 사용을 가능하게 만들어 줄 것이다.

[8] https://en.wikipedia.org/wiki/Table_of_the_largest_known_graphs_of_a_given_diameter_and_maximal_degree