
블록스택 탈중앙 컴퓨팅 네트워크 기술 백서 2.0

2019년 5월

For Blockstack PBC

이 백서의 일부는 아래의 피어 리뷰를 완료한 회의 및 잡지에서 이미 기재된 바 있습니다.

- M. Ali, J. Nelson, R. Shea and M. J. Freedman, “*Blockstack: A Global Naming and Storage System Secured by Blockchains*”, 2016 USENIX Annual Technical Conference, Denver, CO, June 2016.
- J. Nelson, M. Ali, R. Shea and M. J. Freedman, “*Extending Existing Blockchains with Virtualchain*”, Workshop on Distributed Cryptocurrencies and Consensus Ledgers, Chicago, IL, July 2016.
- M. Ali, J. Nelson, R. Shea and M. J. Freedman, “*Bootstrapping Trust in Distributed Systems with Blockchains*”, USENIX; login: Issue: Vol. 41, No. 3, Pages 52-58, Fall 2016.

백서 2.0 버전은 2017년에 발표한 백서 1.0 이후 주요 변경 사항을 담고 있습니다. 이전 백서는 아래를 참조하세요.

- M. Ali, R. Shea, J. Nelson and M. J. Freedman, “*Blockstack: A New Internet for Decentralized Applications*”, Whitepaper Version 1.1, Oct 2017.

백서에서 설명하는 시스템과 개념 일부는 아래의 백서 저자의 박사 논문에서도 논의된 바 있습니다.

- M. Ali, *Trust-to-Trust Design of a New Internet*, PhD dissertation, Princeton University, June 2017.
 - J. Nelson, *Wide-area Software-defined Storage*, PhD dissertation, Princeton University, June 2018.
-

면책조항: 블록스택 토큰인 “스택스 토큰” 또는 “스택스”는 미국 델라웨어 유한책임회사 Blockstack Token LLC가 개발 중인 암호 자산이며 웹사이트 주소는 www.stackstoken.com입니다.

백서 발행이 스택스 발행이나 판매, 또는 스택스 구매를 위한 메커니즘을 의미하지 않습니다. 최종 확정된 스택스 토큰 문서가 발행됐을 때, 스택스 토큰 및 관련 도구의 발행 및 판매가 가능합니다.

여기서 소개한 내용은 미국 1933년 개정 증권법 규정 A에 따라 “사전 시도(testing the waters)”로 간주될 수 있습니다. 우리는 규정 A에 따라 토큰 발행을 완료할 의무가 없습니다. 토큰에 관심있는 투자자 전체가 아닌 일부에게 발행할 수 있으며, 규정 A에 따라 발행하지 않을 수도 있습니다. 미국 증권거래위원회(SEC)가 토큰 발행 성명서(offering statement)를 승인한 후에 토큰 판매가 가능합니다. 성명서에 포함된 정보는 현재 제공하는 정보보다 더 완전할 것이며, 중대한 차이가 발생할 수도 있습니다. 투자하기 전 SEC에 제출한 서류를 반드시 읽어보기 바랍니다.

금전 및 향응을 요구하지 않으며 혹여 그러한 제안이 있어도 절대 수락하지 않습니다. 미국 증권거래위원회(SEC)가 토큰 발행 성명서를 승인할 때까지 주식 매수 제안을 받아들일 수 없으며, 토큰 발행의 구매 금액의 일부도 받을 수 없습니다. 성명서가 승인된 이후, 매수 제안은 수락 통보 이전에는 언제든지 이행 의무나 책임 없이 철회 또는 취소할 수 있습니다.

토큰 발행에 대한 관심의 표시에는 그 어떤 의무 또는 책임이 따르지 않습니다.

스택스 토큰 발행에 관심있는 투자자는 공시 자료 및 토큰 발행 성명서를 검토해야하며 www.sec.gov 에서 사본을 확인할 수 있습니다.

블록스택은 미국 증권거래위원회(SEC), 금융산업규제기구(FINRA) 또는 기타 금융 감독 당국으로부터 증권 거래업 또는 투자 자문업으로 등록, 허가 또는 감독받지 않았으며, 재정적 자문이나 서비스 제공을 허가 받지 않았습니다.

- 백서 한국어본은 영어 원본을 번역한 것이며, 영어 원본과 번역본간 내용 상이성이 발견된다면, 영어 원본이 우선합니다.

블록스택 탈중앙 컴퓨팅 네트워크

Muneeb Ali

Jude Nelson Aaron Blankstein Ryan Shea

Michael J. Freedman^{*}

<https://blockstack.org>

백서 버전 2.0.4

2019년 5월 20일

개요

인터넷 앱의 전통적인 클라이언트/서버 모델과 클라우드 컴퓨팅은 일반적으로 더 많은 서버를 추가함으로써 앱이 확장 가능했습니다. 최근에는 대형 기술 회사에 모든 유저 데이터를 저장하면서 대량 데이터 유출, 단일 지점 실패 및 단일 지점 제어, 유저 프라이버시 침해 등과 같은 몇 가지 문제가 발생했습니다.

본 백서에서는 블록스택 탈중앙 컴퓨팅 네트워크를 소개합니다. 블록스택은 안전한 개인 앱 구축을 위해 기존 클라우드 컴퓨팅에 풀 스택(full-stack) 대안을 제공합니다. 전통적인 인터넷 앱과 다른 블록스택 탈중앙 앱의 주요 특징은 대부분의 비즈니스 로직과 데이터 처리가 앱 공급자가 호스팅하는 중앙 집중식 서버가 아닌 클라이언트 측에서 이루어진다는 점입니다. 탈중앙 컴퓨팅으로의 전환은 여러 측면에서 1980년대의 메인 프레임에서 데스크톱 컴퓨팅으로의 전환과 유사합니다.

블록스택은 네트워크의 핵심을 최대한 단순하게 유지하면서 복잡성은 에지(edge, 유저 기기 및 유저 제어 스토리지)로 넘기는 종단간(end-to-end) 설계 원칙을 따릅니다. 우리 네트워크의 기반은 스택스 블록체인으로서, (a) 탈중앙 앱을 확장하고, (b) 개발자가 네트워크에서 고품질 앱을 개발하면 보상하도록 설계되었습니다. 스택스 블록체인은 새로운 조정가능한 증명(Tunable Proofs) 선거 시스템을 사용하여, 새롭게 생성된 블록체인을 안전하게 부트스트랩(bootstrap)합니다. 새로운 스마트 컨트랙트 언어인 Clarity는 스마트 컨트랙트의 보안 및 예측 가능성을 최적화하고 모든 트랜잭션에 대한 정적 분석을 허용합니다.

우리 아키텍처의 핵심 구성 요소는 확장성과 성능이 뛰어난 탈중앙 스토리지 시스템인 가이아(Gaia)로서, 유저 제어 가능한 개인 데이터 보관함(locker)입니다. 유저는 개인 데이터 저장소를 블록스택 클라이언트 소프트웨어에 연결하고, 앱은 유저 데이터를 데이터 보관함에 직접 기록합니다. 블록스택 Auth라고 하는 범용 ID 및 인증 시스템을 통해 유저는 앱별로 가입할 필요도 없고, 암호 인증보다 안정성이 낮은 비밀번호 입력 기반의 로그인을 할 필요도 없어집니다.

SDK 및 개발자 도구를 사용하면 블록스택 앱 개발이 기존 인터넷 앱 만큼 쉽고, 개발자는 서버 또는 데이터베이스 운영을 걱정할 필요가 없습니다. 블록스택은 2019 년 초에 100 개가 넘는 독립 앱을 제작해 현재까지 운영하고 있습니다. 블록스택의 아키텍처는 지난 수년 간의 제작 경험과 앱 개발자의 피드백을 통해 진화해 왔습니다. 이 백서는 2017년에 발행한 백서의 주요 개정을 담았습니다.

^{*} 프린스턴 대학교 컴퓨터 과학과 교수 및 Blockstack PBC 기술 자문관

1 들어가며

40년 전에 설계된 인터넷은 단순히 연구 프로젝트에 지나지 않았지만 현재 거의 모든 디지털 상호작용 다룰 정도로 성장했습니다. 1990년대 이후, 핵심적인 하위 계층 인터넷 프로토콜은 거의 변화가 없었지만, 인터넷 앱 계층과 서버 인프라는 인터넷 앱의 대규모 성장 지원을 위해 상당한 발전을 이뤘습니다.

인터넷 앱 구축의 기본 모델은 90년대에 대중화 된 클라이언트/서버 모델 [1]입니다. 이 모델은 장기적으로는 부정적인 결과를 낳은 단기적인 축복이었습니다. 이 모델로 인해 웹 환경이 꽃 피웠지만, 웹 서비스가 원격 서버에 상당히 의존하도록 만들었습니다. 클라우드 컴퓨팅은 기본 클라이언트/서버 모델에서 발전한 것으로, 오늘날 클라우드 업체는 개인 유저 데이터를 저장하고, 앱 로직 및 계산을 실행하며, 액세스 자격 증명을 관리하는 등의 작업을 수행합니다.

지난 10년 동안 클라우드 컴퓨팅이 야기한 부정적인 결과를 목도하였고, 클라이언트/서버 모델에 의존하면서 소프트웨어 구축 전체 모델에 대한 의문을 가지게 되었습니다. 대량 데이터 유출[2], 유저 프라이버시 침해[3], 데이터 이식성의 부재, 기술 대기업에 향한 만연한 불신[4] 등이 클라이언트/서버 모델의 핵심 설계에서 비롯됐습니다. 인류 사회에서 컴퓨팅의 중요성이 커짐에 따라 우리는 구형 컴퓨팅 모델이 우리의 삶의 방식을 정의하도록 내버려 둘 수 없습니다.

차세대 클라우드 컴퓨팅 진화는 보다 강력한 클라이언트 디바이스, 에지 컴퓨팅(edge computation) 및 글로벌 연결성을 활용하여 중앙 집중식 플랫폼에 대한 의존도를 낮출 것입니다. 탈중앙 컴퓨팅을 위한 진화는 이미 시작되었으며, 이는 메인 프레임과 데스크톱이 등장한 이래로 컴퓨팅 업계에서 가장 중요한 기술적 변화입니다. 탈중앙 컴퓨팅은 소프트

웨어를 구축하고 사용하는 방법을 변화시킬 수 있습니다. 새로운 도구 세트를 개발자에게 제공하고, 소비자가 소프트웨어와 맺는 관계를 변화시킵니다. 즉 소프트웨어는 유저를 보호하며, 유저 이익을 그 무엇보다 중시하여 최적화합니다.

블록스택은 전통적인 클라우드 컴퓨팅에 대한 풀스택 대안을 제공하는 탈중앙 컴퓨팅 네트워크를 설계, 개발 및 발전시키기 위한 오픈소스입니다. 블록스택은 전통적인 인터넷의 응용 계층을 재구성하고 탈중앙 앱을 위한 새로운 네트워크를 제공합니다. 블록스택에 구축된 앱은 유저가 직접 데이터를 소유하고 제어할 수도록 합니다[5]. 블록스택은 기존 인터넷 전송 계층과 기저 통신 프로토콜을 사용하는 동시에 앱 중앙집중식 지점을 제거합니다. 블록스택은 네트워크의 핵심을 최대한 단순하게 유지하면서 복잡성은 고객에게 넘기는 종단간(end-to-end) 설계 원칙[6, 7]을 따릅니다. 앱 확장을 위해 전세계 앱 상태 변경을 최소화하고, 클라우드 저장소 성능에 필적하는 신뢰성 높은 탈중앙 저장 시스템을 제공합니다. 또한 풀스택 방식은 탈중앙 앱 구축에 필요한 스택요소를 모든 개발자에게 디폴트 옵션으로 제공합니다. 블록스택은 모듈식이며, 개발자는 쉽게 개별 맞춤형 제작과 대체 기술 통합을 할 수 있습니다.

이 백서는 2017년에 발표한 백서의 주요 변경 사항을 담고 있으며 핵심, 제작 및 구축으로 부터 얻은 교훈과 앱 개발자의 피드백을 통해 설계에 반영했습니다.

2016년 피어 리뷰를 마친 백서의 일부분[8, 9, 10]도 시대에 뒤떨어지므로, 최신 블록스택 설계를 담은 이번 백서를 참조하시기 바랍니다. 이번 백서에는, 2장 탈중앙 앱 확장, 개발자가 고품질 앱 구축 시 보상을 제공하도록 설계된 새로운 스택스 블록체인 설계, 3장 보안 및 예측 가능성을 최적화하는 새로운 스마트 컨트랙트 언어, 4장 가이아(Gaia) 탈중앙 스토리지 시스템, 5장 인증 프로토콜, 6장 개발자 도구, 7장 현재 앱 개발자의 블록스택 이용 방식을 강조하여 담았습니다.

1.1 탈중앙 컴퓨팅 개요

탈중앙 시스템이란 기저 인프라를 제어하는 단일 요소가 없는 특별한 유형의 분산 시스템이며, 노드가 네트워크에 참여하면 경제적 보상을 얻습니다. 탈중앙 네트워크에 대한 최근의 관심은 비트코인 백서[11]의 출간과 함께 시작되었습니다. 블록체인과 암호화폐는 현대 탈중앙 시스템에서 중추적인 역할을 합니다. 블록체인과 암호화폐에 관한 배경[12]을 살펴볼길 바랍니다.

현재 다양한 탈중앙 시스템이 제작되고 있습니다. 최초이며 현재 가장 큰 규모의 블록체인 네트워크인 비트코인의 중요한 목표는 비트코인 디지털 통화를 추적하고 소유권 문제를 해결하는 것입니다. 이더리움[13]의 목표는 비트코인 보다 더 일반적입니다. 즉, 스마트 컨트랙트와 탈중앙 앱이 가능한 "세계(world) 컴퓨터"를 만드는 것입니다. Filecoin[14]은 탈중앙 파일 호스팅 및 저장을 위한 네트워크 구축을 시도하지만, 블록스택은 탈중앙 컴퓨팅을 위한 풀스택을 구현하여 블록체인 계층이 최소한의 상태와 로직을 처리하는 안전한 개인(private) 앱을 구현하는 데 중점을 둡니다.

1.2 설계 목표

최적화된 블록스택의 설계는 다음과 같은 특징을 가집니다.

1. **사용 편의성.** 탈중앙 앱은 엔드 유저에게 현 인터넷 상 앱만큼 사용이 쉬워야 하며, 오늘날 클라우드 컴퓨팅 상의 앱 개발만큼 쉬워야 합니다.
2. **확장성.** 탈중앙 앱은 인터넷을 사용하는 수억~수십억명의 유저를 지원해야 하므로, 블록체인을 포함한 네트워크는 유저와 앱의 수에 따라 확장 가능해야 합니다.
3. **유저 제어.** 분산 컴퓨팅을 사용하는 앱은 기본적으로 유저가 제어할 수 있어야 합니다. 유저는 앱이 운영하는 서버에 의존하기 보다, 계산 및 저장소 리소스를 제공할 수 있어야 합니다.

이러한 설계 목표 하에, 블록스택은 "헤비" 블록체인과 "세계 컴퓨터" 설계 철학[13, 15, 16, 17]을 바탕으로

현대적인 탈중앙 컴퓨팅 접근 방식과는 차별화된 설계를 채택했습니다.

블록체인 계층에서 최소 로직과 상태: 확장성을 위해 블록스택은 "라이트" 블록체인 계층에서 앱 로직과 데이터를 최소화합니다. 블록체인을 앱 로직 및 스토리지에 사용하는 것은 본질적으로 "오프체인" 접근 방식보다 느립니다. 왜냐하면, 광범위한 네트워크 및 장치에서 상태를 동기화하고 유효성을 검사해야 하므로 작업 처리량에 상당한 제한이 있습니다. 제한 요소는 글로벌 연결성을 위한 기저 대역폭과 일반 네트워크 노드에서의 메모리/스토리지입니다(즉, 물리적 제한 vs. 프로토콜 제한)

로컬 상태 변경 vs. 글로벌 상태 변경: 블록스택 네트워크는 풀스택 접근 방식을 사용하여 블록스택에 구축된 앱 확장이 가능하게 합니다. 앱 상의 상호 작용은 가능할 때마다 글로벌 상태 변경과 로컬 상태 변경을 초래합니다. 따라서, 블록스택의 스토리지 시스템인 가이아(4장 참조) 및 인증 프로토콜(5장 참조)은 우리 네트워크의 중요한 기본 구성 요소입니다. 이를 통해 앱이 유저의 개인 데이터 보관함과 상호 작용하고 블록체인 트랜잭션 없이 유저를 인증할 수 있습니다. 스택스 블록체인은 탈중앙 방식으로 일관되게 글로벌 상태 전환을 조정하는 데만 사용됩니다(예 : 고유한 유저네임 등록).

신뢰할 수 있는 클라우드 형 스토리지 vs. 피어(peer) 스토리지: 블록스택 상 앱은 유저 개인 데이터 보관함을 사용하여 데이터를 유저에게 저장하므로 서버가 유저 데이터 또는 액세스 자격 증명을 저장할 필요가 없습니다. 이 접근 방식을 통해 유저가 데이터를 제어할 뿐만 아니라 개발자에게 복잡성을 줄여줍니다. 즉, 개발자는 더 이상 서버 및 데이터베이스를 운영하거나 유저 대신 클라우드 인프라 요금을 지불할 필요가 없습니다. 또한 P2P 스토리지 고유의 신뢰성 및 성능 관련 문제를 피할 수 있고, 기존 클라우드 저장 업체를 탈중앙 광역 파일 시스템으로 용도를 변경할 수 있습니다. 블록체인의 계층은 사용자의 데이터 보관함에만 포인터를 저장합니다.

개발자를 위한 풀스택 SDK: 블록스택은 탈중앙 앱을 개발하는데 필요한 모든 계층에 "풀스택(full-stack)" 방식을 채택하고 디플로트 옵션을 제공합니다. 개발자 SDK는 작업 시 블록체인 및 기타 기술의 복잡성을 제거합니다. 즉, 앱 개발자는 SDK의 인터페이스를 사용하여 앱을 쉽게 구축할 수 있습니다(6장 참고). 개발자 스택의 다양한 계층은 모듈식이며 필요에 따라 다른 기술과 함께 사용할 수 있습니다.

탈중앙 컴퓨팅 접근법과의 이러한 차이점 외에도 스마트 컨트랙트 언어는 고유한 설계 결정을 내려 스마트 컨트랙트의 보안 및 예측 가능성을 최적화합니다(자세한 내용은 3장 참조).

1.3 앱을 위한 새로운 모델

블록스택은 개발자에게 앱 구축을 위한 새로운 모델을 제공하여 기본적으로 앱이 탈중앙화되고, 유저가 제어할 수 있도록 합니다.

- 1. 불투명한 데이터 베이스 없음:** 클라이언트/서버 모델에서, 서버측은 다량의 유저 데이터를 저장하고 쿼리해야 하기 때문에 데이터베이스는 모든 앱의 핵심입니다. 탈중앙 컴퓨팅에서는 개발자가 애초에 데이터를 호스트하지 않기 때문에 데이터베이스를 관리하거나 보안을 유지할 필요가 없습니다. 개발자는 주로 앱 로직에 중점을 둡니다. 즉, 유저는 앱을 다운로드하고 개인 데이터 보관함에 접속(plug-in)합니다. 데이터베이스는 공공 데이터의 색인을 생성하는 기존 인터넷 서비스의 "서치 인덱서(search indexer)"와 기능이 동일합니다. 누구나 기저의 (탈중앙) 데이터를 사용하여 인덱스를 만들 수 있습니다.
- 2. 서버 없음:** 클라이언트/서버 모델에서, 모든 유저의 연산이 서버측에서 실행될 때 앱은 서버를 추가하여 확장 가능합니다. 탈중앙 컴퓨팅에서 앱은 클라이언트측에서 실행되며, 새로운 유저는 (개발자에 의지하기 보다) 각자의 연산 및 저장 용량을 네트워크로 가져옵니다. 각 유저가 앱 사용에 필요한 저장 및 컴퓨팅 리소스를 가져오기 때문에, 개발자는 앱 코드 호스팅을 위한 최소한의 인프라만 제공하면 됩니다.
- 3. 스마트 컨트랙트:** 클라이언트/서버 모델에서 글로벌 상태 변경은 네트워크에서 유일한 진실 판명 권한을 지닌 중앙 서버가 조정합니다. 탈중앙 컴퓨팅에서 상태 변화는 오픈소스 블록체인에서 실행되는 스마트 컨트랙트를 통해 발생합니다.
- 4. 탈중앙 인증:** 전통적인 인터넷에서 유저는 신뢰할 수 있는 인증 프로세스를 사용하여 인증합니다. 앱이 유저 데이터베이스를 유지 관리하는 경우 앱은 암호, 때로는 추가 암호를 사용하여 유저를 인증합니다. 앱이 Google이나 Facebook과 같은 제 3자 신원 인증 서비스를 이용해 인증하면, OAuth [19] 프로토콜을 사용합니다. 물론 이런 접근법은 유저로부터 프로세스 제어를 제거합니다. 탈중앙 컴퓨팅에서 인증은 유저의 클라이언트가 수행하며, 블록체인에 고정된 특정 유저네임에 대한 제어를 증명하는 성명서에 암호화 된 서명을 하면 인증할 수 있습니다. 모든 앱은 이러한 증거를 독립적으로 검증할 수 있습니다.
- 5. 네이티브 토큰(Native tokens):** 전통적인 인터넷 앱에서 대금 지급은 주로 신용카드와 같은 제 3자 서비스를 통해 이루어집니다. 디지털 토큰이란 블록스택과 이더리움과 같은 탈중앙 컴퓨팅 플랫폼의 네이티브 자산입니다. 유저는 네이티브 토큰에 대해 직접적인 소유권을 가지며 디지털 자산 및 스마트 컨트랙트 등록, 스마트 컨트랙트 실행에 필요한 비용을 지불하는데 사용 가능합니다. 네이티브 토큰의 사용은 스마트 컨트랙트를 통해 프로그래밍 가능하고, 구독 서비스 구축과 다른 앱 기능을 자동화 할 수 있습니다. 기존의 인터넷 앱 개발자는 이러한 프로그래밍 가능한 토큰을 만들 수 없었습니다.

1.4 탈중앙 컴퓨팅의 계층

블록스택 탈중앙 컴퓨팅 네트워크는 기존 인터넷 설계로는 "앱 계층"에 논리적으로 존재하지만, 블록스택 네트워크 자체가 다양한 시스템으로 구성되어 있고, 이는 탈중앙 앱 실행에 필요한 컴포넌트를 제공합니다.

- 1. 스택스 블록체인:** 블록스택 네트워크의 토대는 스택스 블록체인으로서 유저가 범용 유저네임과 같은 디지털 자산을 등록/제어하고, 스마트 컨트랙트를 등록/실행할 수 있게 해줍니다. 또한 범용 유저네임과 같은 디지털 자산은 유저가 데이터 저장소를 제어하는 등 많은 기능을 부여합니다. 즉 유저는 개인 데이터 보관함 접근 인증을 범용 유저네임과 연결합니다.
- 2. 가이아(Gaia):** 가이아 스토리지 시스템은 유저 제어 스토리지 시스템으로서 앱과 개인 데이터 보관함이 교류할 수

있도록 합니다. 유저는 이러한 암호화된 데이터 보관함을 클라우드 업체, 로컬 디스크 또는 원격 저장소에서 호스팅할 수 있습니다. 무엇보다, 유저는 공급 업체 선택권을 가지고 있습니다. 가이아의 데이터는 암호화 되어 있고, 유저가 암호키로 클라이언트 측에 서명합니다. 유저 데이터 보관함을 찾으려면 스택스 블록체인에 정보를 검색하면 됩니다.

3. **블록스택 인증:** 블록스택 인증 프로토콜은 앱을 이용하는 탈중앙 인증 프로토콜입니다. 이를 통해 유저는 자신의 ID를 사용하여 인증할 수 있고, 가이아 내 유저의 앱 데이터 저장 장소에 대한 정보를 제공 받습니다.
4. **블록스택 라이브러리**와 **SDK:** 소프트웨어 스택의 최상위 계층에는 개발자 라이브러리 및 SDK가 있으며, 앱 개발자와 유저가 블록스택 네트워크의 다양한 컴포넌트와 상호작용할 수 있습니다. 예를 들어 블록스택 클라이언트 소프트웨어를 사용하면 유저가 자신의 ID를 등록하고 관리할 수 있습니다. 블록스택 개발자 라이브러리를 이용하여 개발자가 기존의 웹상의 앱 구축처럼 쉽게 블록스택 앱을 구축할 수 있습니다.

2. 스택스 블록체인

블록스택 네트워크의 기본(foundation) 계층은 스택스 블록체인입니다. 스택스 블록체인은 네트워크에 글로벌 컨센서스 및 조정 계층을 제공하고, 스택스 토큰이라는 블록스택 네트워크 자체 토큰을 실행합니다. 스택스 토큰은 사용자가 범용 유저네임, 소프트웨어 라이선스, 포인터 등의 디지털 자산을 저장소에 등록할 때 "연료"로 소비됩니다. 스마트 컨트랙트를 등록/실행하는 마이너(채굴자)에게 보상을 할 때에도 사용됩니다.

이번 장에서는 스택스 블록체인의 고 수준(high level) 설계를 소개하겠습니다. 설계의 실행과 발전에 대한 상세한 설명은 다양한 컴포넌트를 다룬 스택스 개선 제안 (SIPs, Stacks Improvement Proposals)¹을 참조하시기 바랍니다. 스택스 개선 과정에서 더 많은 SIP가 수용되면 백서를 업데이트할 예정입니다. 스택스 블록체인은 다음의 설계 결정을 포함하고 있습니다.

1. 리더 선출을 위한 조정가능 증명(Tunable Proofs) 메커니즘
2. 기존 블록체인의 해시파워를 재사용하는 소각증명(Proof-of-Burn) 채굴 알고리즘
3. 피어간 연결을 위해 그래프 랜덤 워크(Random Graph Walk)를 사용하고 합의에 도달하기 위한 데이터 양을 줄인 새로운 피어 네트워크인 아틀라스(Atlas)
- ¹ <https://github.com/blockstack/blockstack-core/tree/develop/sip>를 참조하세요.
4. 튜링 불완전(non-Turing complete)하며 해석되는 스마트 컨트랙트 언어(Clarity)

블록체인 버전: 현재 스택스 블록체인은 기본 기능 배포를 위한 초기 구현 단계인 "Ver. 1"입니다. 스택스 블록체인 Ver. 1은 비트코인 네트워크를 사용하여 합의 알고리즘을 구현하고 전송 작업과 같은 스택스 토큰 운영을 지원합니다. 또한, 블록스택 네이밍 시스템(Blockstack Naming System)[8] 등에 스마트 컨트랙트를 실행합니다. Ver. 1의 실행 및 기능에 대한 자세한 내용은 Github[20] '실행' 탭을 참조하세요. 이번 장의 나머지 부분은 스택스 블록체인의 " Ver. 2" 설계를 다루고 있습니다. 스택스 블록체인 Ver. 2는 새로운 합의 알고리즘 및 스마트 컨트랙트 언어의 모든 기능을 구현하며 Ver. 1과 비교해 상당한 업그레이드가 된 것입니다.

2.1 리더 선출

블록스택의 1세대 블록체인은 논리적으로 Layer-1(L1)의 최상단에서 운영되었고, 각 트랜잭션은 L1 비트코인 트랜잭션과 1대1 대응을 이루었습니다. Namecoin[8]과 같은 더 작은 블록체인 네트워크의 보안 문제로 얻은 교훈을 바탕으로, 블록스택의 블록체인 재구성이 비트코인 블록체인을 재구성하는 것과 난이도가 비슷하도록 하기 위해서입니다.

스택스 블록체인은 리더 선출 프로세스에 조정가능 증명(Tunable Proofs) 메커니즘을 사용합니다. 조정가능 증명 메커니즘은 다양한 메커니즘에서 입력을 받아 각 입력에 주어진 가중치를 적용할 수 있는 리더 선출 시스템입니다. 예컨대, 조정가능 증명을 이용해 네이티브 작업증명 알고리즘에 기능을 추가하여, 더 성공한 블록체인의 해시파워를 재사용할 수 있습니다. 조정가능 증명 마이닝(채굴)은 새 블록체인을 안전하게 부트스트랩하여 서서히 네이티브 작업증명 메커니즘 사용으로 전환하는 것을 목표로 합니다. 조정가능 증명은 (a) 네이티브 작업증명, (b) 다른 암호화폐의 소각증명의 두 부분으로 구성됩니다.

처음에는 마이닝의 소각증명 (proof-of-burn)이 더 큰 비중을 차지합니다. 소각증명을 통해, 마이너는 암호화폐를 소각하여 마이닝 과정 참여에 대한 관심을 표시합니다. 리더로 선출되기 위해, 후보자는 기저 암호화폐(예, 비트코인)를 소각하고, 리더 예정 블록의 초기 트랜잭션에 전념합니다. 이는 리더의 포크 선택의 역할도 합니다. 즉, 블록의 합의 (consensus) 해시에는 이전 블록 헤더가 포함되어야 합니다. 다수의 경쟁 포크가 있는 경우, 패배한 포크에서 "채굴"을 선택한 리더는 블록 보상, 트랜잭션 수수료를 받지 못하거나 소각된 암호화폐를 회복시키지 못합니다.

스택스 블록체인에 사용되는 소각 검증 메커니즘은 다음의 특징을 가집니다.

높은 검증 처리량. 처리된 스택스 트랜잭션의 수는 기저의 "소각 체인" (예, 비트코인)의 트랜잭션 처리 속도와 분리되어 (de-coupled) 있습니다. 소각 증명 선출을 사용하면 스택스 트랜잭션의 모든 블록이 기저 소각 체인에 있는 새 블록을 확인할 수 있습니다.

저지연(low-latency) 블록 포함. 소각증명 합의 알고리즘을 통해 단일 리더 선출이 가능하므로, 현재 리더는 댄풀 (Mempool)의 새로운 트랜잭션을 즉시 스택스 블록에 포함할 수 있습니다. 블록 스트리밍 모델을 사용하면 블록이 수 초 안에 트랜잭션을 포함할 수 있습니다.

공개 리더십 세트. 소각증명 선거를 통해 누구나 리더가 될 수 있습니다. 이 메커니즘 통해 스택스 블록체인이 공개 블록 체인이 될 수 있습니다(정해진 리더 또 위임된 지분증명 시스템에 의존하여 폐쇄적으로 기능하는 블록체인과 대비됨). 또한, 합의 알고리즘은 단일 리더 선거를 수행하여 예비 리더는 조정 역할을 할 필요가 없습니다.

채굴기(mining hardware) 없는 참여. 리더로 참여하려면 전통적인 작업증명 마이닝 계획보다 암호화폐 소각이 필요합니다. 따라서, 리더로 참여하기 위해서는 채굴기가 필요하지 않으며, 비용 부담 능력이 작더라도 소각할 암호화폐를 획득할 수 있다면 누구든지 마이닝에 참여할 수 있습니다.

공정한 마이닝 풀(pool). 스택스 블록체인은 기본적으로 공정한 마이닝 풀을 지원합니다. 네트워크에 참여하는 누구든지 암호화폐를 소각하여 리더 선출을 할 수 있습니다. "유저 지원 소각(user support burns)"을 실행한 유저는 리더와 동등한 비율로 스택스 블록의 보상을 받습니다.

대체 작동(failover) 기능. 대체 작동 설계 덕분에, 소각 체인이 불안정하거나 스택스 체인을 마이닝하기에 부적합할 경우 스택스 체인은 다른 소각 체인을 사용할 수 있습니다. 소각증명 요소에 대한 자세한 내용은 [21]을 참조하세요. 향후 스택스 블록체인에 네이티브 해시파워가 충분해지면, 소각증명(Proof-of-Burn) 컴포넌트가 필요하지 않을 수도 있습니다.

2.2 조정가능 증명

스택스 블록체인은 소각증명 뿐만 아니라 네이티브 작업증명 컴포넌트를 합의 알고리즘에 포함하고 있습니다. 이 조합을 통해 체인의 보안 책임을 SIP-001에서 설명한 소각증명 선거 시스템과 공유할 수 있습니다. 이 네이티브 작업증명과 소각증명의 결합을 조정가능 증명(Tunable Proofs)이라고 명하며, 소각증명을 통해 체인 안정성이 보장되는 네이티브 작업증명 마이닝의 책임있는 도입이 가능합니다. 작업증명 이윤이 낮을 때에도 가능합니다. 조정가능성이 기저 소각 체인의 가치 악화 시 마이그레이션에 대한 가능성을 열었습니다. 조정가능 증명을 통해 다른 작업증명이나 지분증명(proof-of-stake) 메커니즘을 연구하고 조정가능한 방식으로 이를 점진적으로 도입가능케 합니다.

리더 선출에서 작업증명 컴포넌트는 리더 후보가 작업증명 논스(nonce)를 소각 트랜잭션에 옵션으로 포함하도록 합니다. 논스 생성에 필요한 작업량(즉, 결과 해시의 소수점 이후 0의 수 i.e. some function the number of leading zeroes in the resulting hash)은 후보자의 "소각량"에 포함됩니다. 처음에 네이티브 작업증명(Proof-of-Work)에 5% 상한선이 적용될 것입니다(제출된 소각량과 비례함). 네이티브 작업증명 컴포넌트는 계속해서 활발히 개발 및 설계 중이며, 보다 구체적인 내용이 발표되면 다 이 장(과 해당 SIP)을 업데이트 하겠습니다.

2.3 아틀라스 피어 네트워크(Atlas Peer Network)

아틀라스 피어 네트워크는 콘텐츠 주소 지정이 가능한 피어 네트워크로서, 각 피어가 네트워크 상의 다른 피어를 추적하고, 모든 데이터의 전체 복사본을 저장하는 가십 프로토콜을 구현합니다. 네트워크의 용량은 스택스 블록체인의 이해 속도가 제한됩니다. 즉, 데이터 세트에 새로운 것이 가입되면 스택스 블록체인의 트랜잭션과 연결되어야 합니다. 아틀라스 피어 네트워크는 스택스 블록체인의 서버 시스템으로 작동합니다. 비구조화된 피어 네트워크로 설계되어 네트워크에 합류하거나 떠나는 노드와 관련된 문제를 피할 수 있습니다[18, 22]. 또한 모든 노드가 모든 데이터의 복사본을 보관하고, 데이터 인덱스는 스택스 블록체인의에서 찾을 수 있으므로, 새로운 아틀라스 노드는 저장해야 할 데이터와 신속하게 동기화할 수 있습니다. 이는 다른 피어에 저장해야 하는 데이터가 무엇인지 사전에 알고 있기 때문에 가능합니다(일반적으로 P2P 네트워크에서는 노드에게 제공되지 않음). 아틀라스 네트워크는 스택스 블록체인의 "확장 스토리지" 서버 시스템 기능을 합니다. 우리는 스택스 블록체인과 직접적으로 상호작용하고 스택스 블록체인 상의 데이터 저장을 최소화하여 설계하고자 합니다. BNS (블록스택 네이밍 시스템) 스마트 컨트랙트[8]와 같은 블록스택 상의 다양한 앱은 변경 불가능(immutable)하며 시간이 기록된(timestamped) 데이터를 저장하는 메커니즘이 반드시 필요합니다. 이 메커니즘은 BNS에서 해당 유저 프로필 및 앱 데이터를 검색에 이용되는 라우팅 정보와 유저네임을 연결하는데 사용됩니다. 대부분의 경우 데이터는 블록체인 자체에 직접 저장되지만, 블록스택은 블록체인의 해시를 저장하고(고비용), 각 해시에 해당하는 데이터를 교환하기 위해 별도의 피어링 네트워크를 구현했습니다.

2.4 스택스 토큰 사용

스택스 블록체인에 의해 구현된 네이티브 스택스 토큰을 통해 블록스택 네트워크에서 다양한 기초 작업이 가능합니다.

1. **디지털 자산 등록을 위한 연료.** 스택스 토큰은 유저네임, 도메인 네임, 소프트웨어 라이선스, 팟캐스트 등 다양한 디지털 자산을 등록하는 데 사용됩니다.
2. **스마트 컨트랙트 등록 및 실행을 위한 연료.** 스마트 컨트랙트를 실행하기 위해서는 스마트 컨트랙트의 정확성 검증 및 실행을 위한 연료가 필요합니다. 스택스 토큰은 스마트 컨트랙트를 스택스 블록체인에 저장하는 비용으로 사용됩니다.
3. **트랜잭션 수수료.** 스택스 토큰은 스택스 블록체인상의 트랜잭션 등 다양한 트랜잭션 수수료 지불에 사용됩니다.
4. **고정된(anchored)앱 체인.** 블록스택에서 폭발적 인기를 끄는 앱을 위해, 블록스택 블록체인은 앱이 블록체인과 스택스 블록체인을 초기 설정할 수 있는 확장성을 위한 진입 경로(on-ramp)가 있습니다. 이러한 “앱 체인”은 스택스를 소각하여 마이닝 및 발전을 위해 사용합니다.

위의 목록은 완성 본이 아닙니다. 블록스택 네트워크가 발전해가면서 네트워크 참여자가 스택스 토큰의 용처를 더 많이 발견할 것이기 때문입니다. 토큰 보유자가 “앱 마이닝”이라는 개발자 보상 프로그램에 참여할 수 있는 “앱 스테이킹” 메커니즘 연구가 현재 활발히 진행 중입니다[23].

3. 스마트 컨트랙트 언어

스택스 블록체인은 디지털 자산을 프로그래밍 방식으로 제어하기 위한 스마트 컨트랙트의 출시 및 실행을 지원합니다. Clarity라는 새로운 스마트 컨트랙트 언어는 보안 및 예측 가능성을 최적화하여 이전 시스템과 차별화되는 몇 가지 주요 설계 목표를 지닙니다.

1. 컨트랙트 언어는 런타임과 공간 요구량에 대해 빠르고 정확한 정적 분석을 허용해야 합니다. 이를 지원하기 위해, 언어는 단일 트랜잭션 실행에서는 튜링 불완전(non-Turing complete)이지만, 트랜잭션 전체 기록을 고려하면 언어는 튜링 완전(turing complete)입니다.
2. 스마트 컨트랙트는 컴파일되는 것이 아니라, VM에 의해 인터프리트 됩니다. 개발자가 작성한 코드는 블록체인 위해 직접 설치(deploy)되어야 합니다.

위 두가지 특성을 만족하기 위해, 스마트 컨트랙트 작성을 위해 특별히 고안한 변형된 LISP가 작성되었습니다. Clarity의 설계에 대한 자세한 설명은 SIP-002 [24]를 참조하세요.

1. 언어 개요

Clarity는 다른 변형된 LISP(예: scheme)와 유사하지만 차이점은 다음과 같습니다.

1. 재귀호출은 불법이고, 람다(lambda)함수는 존재하지 않습니다.
2. 루핑(Looping)은 맵, 필터, 폴드를 통해서만 수행 가능합니다.
3. 유일한 Atomic 타입은 부울식(booleans), 정수, 고정 길이 버퍼 및 기타 주요 타입들
4. 다양한 Atomic 타입에 대한 추가로 지원하지만, Clarity에서 유일한 가변 길이 목록은 함수 인풋으로 나타냅니다 (즉, 추가 또는 결합과 같은 목록 연산에 대한 지원이 없음). 각 값마다 이름 붙이고 타입하는 튜플(named-and-typed tuples)도 지원합니다.
5. 변수는 let 바인딩을 통해 생성될 수 있으며, set와 같은 함수 변경을 지원하지 않습니다.
6. 상수 및 함수 정의는 코드를 단순화하기 위해 서술문을 통해 정의 가능하지만, 이는 순전히 문법적(syntax)입니다. 즉, 정의가 즉시 처리(inline) 안되면 컨트랙트는 불법으로 간주되어 거절됩니다. 정의는 또한 사적(private)이기도 한데, 이 방식으로 정의된 함수는 주어진 스마트 컨트랙트에 정의된 다른 함수에 의해서만 호출될 수 있습니다.
7. define-public 문을 통해 정의된 함수는 public 함수입니다. 이 함수의 인자는 타입을 지정해야 합니다.

스마트 컨트랙트는 다음과 같은 특징도 있습니다.

1. 다른 스마트 컨트랙트에서 public 함수 호출하기. 이러한 스마트 컨트랙트는 해시로 식별되며, 호출하는 스마트 컨트랙트가 체결될 때 이미 존재해야 합니다. 이는 재귀도 불법 처리하고, 기존 스마트 컨트랙트 플랫폼에 있는 공통 공격 벡터인 함수 재진입도 막습니다.
2. 디지털 재산 소유 및 제어. 스마트 컨트랙트는 공개키나 멀티 서명 주소처럼 최우선 주체입니다.

각 스마트 컨트랙트는 자체 데이터 공간이 있습니다. 데이터 공간 내의 데이터는 맵에 저장됩니다. 이러한 저장공간에서 형식화된 튜플(typed-tuple)끼리 연결됩니다(형식화된 키-밸류 저장소와 거의 유사). 테이블 데이터 구조와는 대

조적으로, 맵은 주어진 키가 정확히 하나의 값을 가집니다(key-value).

모든 스마트 컨트랙트는 다른 스마트 컨트랙트 맵에서 데이터를 가져올 수 있지만, 자신의 스마트 컨트랙트만 자체 맵 내의 데이터를 직접 업데이트할 수 있습니다.

다른 데이터 구조와는 달리 데이터 맵을 사용하기로 한 이유는 다음과 같습니다.

1. 데이터 맵의 단순성을 통해 VM 내에서 단순한 구현과 쉬운 함수 추론이 가능합니다. 주어진 함수 정의를 검사함으로써 어떤 맵이 수정될지, 맵 내에서 어떤 키가 호출의 영향을 받는지 명확하게 알 수 있습니다.
2. 데이터 맵 인터페이스는 맵 연산의 리턴 타입이 고정 길이라는 것을 보장합니다. 이는 스마트 컨트랙트의 런타임, 비용 및 기타 속성의 정적 분석에 필요한 사항입니다.

3.0.2 튜링 불완전과 정적 분석

튜링 불완전(non-Turing complete)언어를 만드는 것이 설계시 중요한 고려 사항이었습니다. 블록체인의 불리한 환경에서 튜링 불완전을 이용하면 프로그래밍 할 때 많은 이점이 있습니다.

1. 튜링 불완전을 통해 정적 분석이 주어진 트랜잭션 실행 비용을 결정하고, 네트워크가 주어진 거래에 부과되는 수수료를 선형적으로 알 수 있습니다. 클라이언트가 트랜잭션 안내(broadcasting) 비용을 잘 알고 있고, 유저에게 비용이 쉽게 전달될 수 있으므로 클라이언트의 행동이 개선됩니다.
2. 튜링 불완전을 통해 정적 분석이 ‘단일 트랜잭션이 호출할 수 있는 다른 컨트랙트’와 같은 중요한 특성을 신속하게 판단할 수 있습니다. 클라이언트가 유저에 주어진 트랜잭션에 대한 부작용에 대해 경고할 수 있으므로 유저 경험에 향상됩니다.
3. 향상되고 정확한 정적인 분석을 통해 프로그래머는 스마트 컨트랙트 발행 전 발생할 수 있는 오류 및 실수에 대해 자신있게 분석할 수 있습니다.

근본적으로 스마트 컨트랙트 프로그래밍을 다른 형태의 프로그래밍과 똑같이 여기면 안됩니다. 블록체인의 특성으로 인해 스마트 컨트랙트의 구성 요소가 더 중요해집니다. 스마트 컨트랙트에 대한 인간과 기계의 이해를 높이기 위해 비슷한 프로그래밍과 비교하는 것은 좋은 방법입니다. 기존의 스마트 컨트랙트의 실제 사용 결과, ‘튜링 완전 스마트 컨트랙트의 역사’는 사실 ‘스마트 컨트랙트 버그의 역사’와 같습니다.

Clarity에서 스마트 컨트랙트가 다음과 같은 정보를 제공할 수 있음을 알리기 전에 정적분석을 실시해야 합니다.

1. 주어진 트랜잭션을 입력 크기의 함수로 안내(broadcasting)하는 비용
2. 특정 표를 수정할 수 있는 트랜잭션 세트

향후 노력을 통해 스마트 컨트랙트 코드를 자동 교정하는 능력 등 고급 분석기능을 지원할 수 있을 것입니다.

3.0.3 인터프리터 언어 vs. 컴파일

Clarity의 두 번째 주요 설계 결정은 컴파일 된 언어가 아닌 인터프리터 언어를 선택한 것입니다(예, WASM으로 컴파일). 컴파일러를 사용하지 않기로 한 우리의 설계 결정은 현대적인 접근 방식과 근본적으로 차이가 있습니다. 이 설계 결정의 주된 이유는 구현 버그를 추론할 수 있기 때문입니다.

구현 버그는 실제로 존재하며, 최고의 코딩 표준을 사용하더라도 피할 수 없습니다. 이는 다른 코드와 스마트 컨트랙트 버그(블록체인)에도 적용됩니다.

스마트 컨트랙트 버그는 처리하기가 더 복잡합니다. 다양한 블록체인 커뮤니티는 블록체인이야말로 궁극적인 진실의 근원이라며 “코드=법” 철학 및 기조를 준수하고 있습니다. 스마트 컨트랙트를 쓰는 개발자는 소스 코드를 통해 자신의 의도를 표현하지만, 컴파일러는 컴파일 과정에서 개발자 의도를 실제 규칙으로 번역합니다. 따라서 컴파일러 버그로 인해 실제 규칙과 개발자 의도가 다른 경우가 발생합니다. 개발자 의도와 규칙 중 어느 것이 더 중요한지에 대한 논쟁을 하는 불쾌한 상황을 초래되기도 합니다. 이러한 상황을 피하기 위해, 스택스 블록체인은 컴파일 단계를 제거하고 개발자의 의도를 블록 체인에 직접 위임하여 개발자의 의도가 규칙에서 벗어나지 않도록 합니다.

스마트 컨트랙트 구현시 버그를 생각해 봅시다. 스마트 컨트랙트 언어가 인터프리터를 사용하는 경우 버그 수정은 상대적으로 적용하기 쉽습니다. 전 세계 모든 계약 코드는 블록체인 자체에 있으며, 단지 인터프리터에 버그 수정을 적용하고 블록체인을 다시 제네시스 블록부터 시작하면 됩니다(모든 트랜잭션을 재적용).

그러나 스마트 컨트랙트 언어가 컴파일되고, 버그가 VM이 아닌 컴파일러에 있다면 구제 방법이 훨씬 더 모호하므로 논쟁의 여지가 더 큼니다. 컴파일러의 버그로 인해 생성된 코드 (결국 블록체인에서 안내되는 코드)가 개발자의 의도된 코드와 다르게 작동할 수 있기 때문입니다. “코드=법” 철학이 당면시 되는 암호화폐 커뮤니티에서 이 상황은 더욱 복잡합니다.

개발자가 작성한 코드는 정확하지만 블록체인 자체에서 생성된 트랜잭션은 잘못된 경우, 모든 개발자의 소스 코드를 수집하고 다시 컴파일하는 것은 현실적 방법이 아닙니다. 특히 소스 코드가 변경되지 않았는지 확인할 수 없는 경우에는 더

혹 비현실적 입니다. 실제로, 그러한 상황에서 블록체인에 게시된 코드는 대부분의 경우 진실의 궁극적 소스(ultimate source of truth)입니다. 어떤 경우든지 간에 개발자는 소스 코드가 아닌 코드를 추론하고 검증해야 합니다. 스마트 컨트랙트가 정확하게 실시되기 위해서 높은 수준의 인터프리터 언어를 사용하는 것이 가장 중요하다고 생각합니다.

4. 가이아: 유저 제어 스토리지

블록스택은 유저가 개인 데이터 락커와 상호작용 할수있는 가이아 스토리지 시스템인 가이아를 이용해 자신의 데이터를 제어할 수 있도록 합니다. 가이아는 유저 제어 스토리지 시스템으로, 앱과 개인 데이터 보관소와 교류할 수 있도록 합니다. 데이터 보관소는 클라우드 또는 다른 데이터 저장 업체가 호스팅할 수 있습니다. 무엇보다, 유저는 어떤 업체가 사용되는지 제어할 수 있습니다. 가이아 상의 데이터는 암호화 되어 있으며 유저가 제어하는 암호키로 서명합니다. 이론상으로, 가이아는 파일 저장을 위해 광역 파일시스템으로 작동합니다.

유저는 가이아 스토리지 시스템을 이용해 데이터를 저장 위치를 지정할 수 있습니다. 가이아 저장 위치에 대한 “포인터”만 스택스 블록체인(과 아틀라스 서브 시스템)에 저장됩니다. 유저가 애플리케이션 및 서비스에 로그인 할 때 블록스택 인증 프로토콜 (5절 참조)은 애플리케이션에 해당 포인터를 전달합니다. 이 정보를 가지고, 앱을 특정 가이아 저장 소와 소통하는 방법을 알려 앱 데이터는 유저가 명시한 저장소에 저장됩니다

가이아의 설계 철학은 최종 유저가 기저 클라우드 업체를 신뢰할 필요 없이 기존 클라우드 업체 및 인프라를 재사용하는 것입니다. 우리는 클라우드 스토리지 업체(Amazon S3, Google 클라우드 스토리지, 또는 로컬디스크)를 “명칭한 드라이브”로 여기고, 이곳에 암호화되고(거나) 서명한 데이터를 저장합니다. 클라우드 업체는 유저 데이터를 볼 수 없고, 데이터의 암호화된 형태(blob)만 볼 수 있습니다.

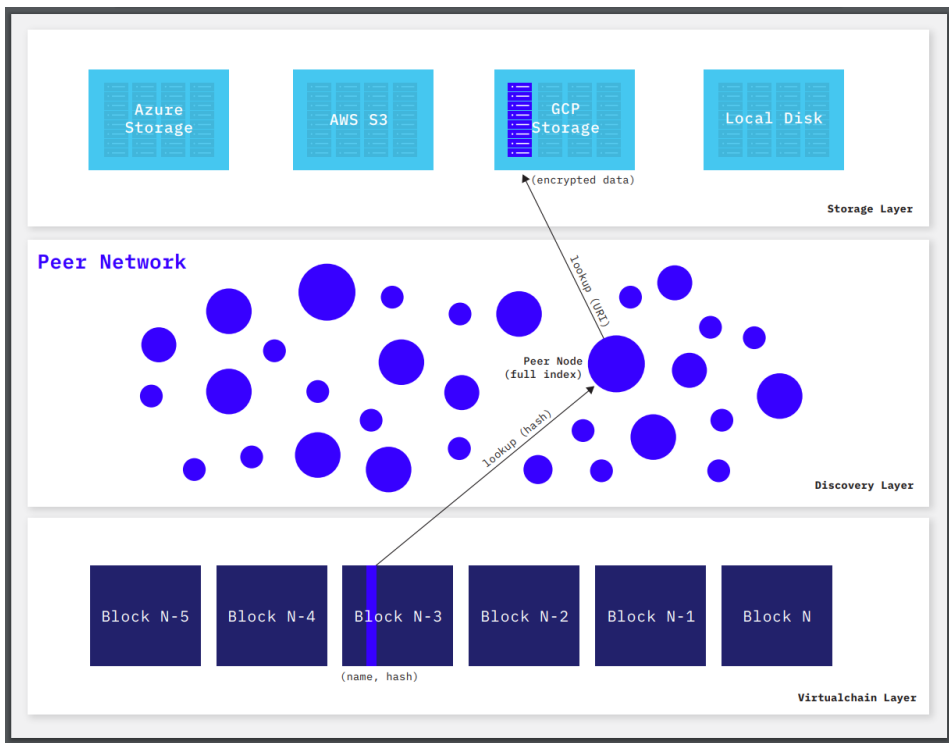


그림 1: 가이아 개요 및 데이터 검색 단계

또한, 관련 공개키나 데이터 해시를 스택스 블록체인을 통해 발견할 수 있으므로 클라우드 업체는 유저 데이터를 위변조할 수 없습니다.

가이아 서버에 데이터를 쓰는 것은 서버의 적절한 위치에 포스팅 하는 것도 포함합니다. 이런 쓰기 요청이 서명된 인증 토큰을 수반하는지 체크함으로써 서버가 포스트를 검증합니다. 토큰은 개인키로 서명하며, 이 키는 데이터를 쓰는 특정 버킷을 제어합니다. 각각의 앱에 분리된 버킷을 제공하기 위해 유저는 분리된 개인키를 앱에서 제거합니다. 개인키는 가이아 서버에서 특정 버킷에만 접근을 허용합니다.

리키는 URL을 포함합니다. 이 서명된 JSON object는 유저의 가이아 데이터 저장소를 가리키는 URL을 포함합니다. 일단 앱이 유저의 가이아 데이터 저장소 위치를 알면 일반 HTTP 요청을 이용해 파일을 요청합니다. 다른 유저가 생성한 파일을 검색하기 위해 앱은 이러한 절차에 따른 검색을 온전히 클라이언트 측에서 수행할 수 있습니다. 이는 초기 검색에서 시간 지연을 초래하지만 라우팅 정보의 많은 부분이 브라우저(또는 네이티브 앱)에 의해 로컬 캐시로 처리되며, 이에 따른 검색은 인터넷상의 기존의 데이터 가져오기(fetch)만큼 빠릅니다.

그림 1은 가이아의 개요를 보여줍니다. werner.id와 같은 이름 데이터 검색을 하면 다음과 같은 순서를 거칩니다.

1. 이름-해시 쌍을 얻기 위해 스택스 블록체인에서 이름을 검색합니다.
2. 이름의 라우팅 정보 파일을 얻기 위해 블록스택 아틀라스 피어 네트워크에서 해시(이름)를 검색합니다.
3. 유저의 가이아 URL을 라우팅 파일에서 얻고 스토리지 백엔드(backend)를 연결하기 위해 URL을 검색합니다.
4. 지정된 가이아 서비스에서 데이터를 얻거나 입력하고(데이터를 읽으려는 사람이 접근 권한이 있거나 필요시에는 암호 해독) 각각의 서명 또는 해시를 검증합니다.

1, 2번에서 /v1/names/<name>엔드포인트에서 블록스택-코어를 단일 호출합니다. 반복적인 데이터

읽기 및 쓰기는 개발자 라이브러리에서 자동적으로 처리됩니다.

성능. 블록스택 아키텍처의 목표는 클라우드 업체에 구축된 기존 인터넷 앱에 필적하는 성능을 제공하는 것 입니다. 통제 중앙 지점 및 실패 중앙 지점을 제거하여 의미있는 보안성과 결합 내결함성을 도입했습니다. 일반 유저 기준으로 오버헤드(overhead)가 중요하거나 두드러지지 않는다면 읽기 및 쓰기 성능 향상을 위해 약간의 오버헤드를 추가할 가치가 있습니다. 가이아(Gaia)의 읽기 및 쓰기 성능을 평가한 결과 기저 스토리지를 이용해 경쟁력있는 속도로 파일을 읽고 쓴다는 것을 입증했습니다. 가이아는 암호화(약 5 % 증가)로 인해 파일 당 무시할 수 있는 정도의 일정한 저장 공간이 추가됩니다. 암호화를 위한 CPU 오버헤드가 있지만 파일 크기 차이가 매우 작기 때문에 읽기 및 쓰기를 위한 네트워크 성능은 기저 스토리지 서비스에 직접 액세스하는 것과 유사합니다.

시스템 확장성. 아키텍처의 스토리지 계층은 확장성에 병목 역할을 하지 않습니다. 현대 클라우드 스토리지 시스템은 확장성이 뛰어나다[25]. 아틀라스 네트워크 또한 개인 유저 파일이나 파일 체크를 인덱스하지는 않지만, 유저의 스토리지 백엔드에 포인터를 인덱스하므로 확장성이 뛰어나다. 스토리지 백엔드는 대량의 데이터 읽기/쓰기를 처리하며, 아틀라스 네트워크는 (a) 사용자가 스토리지 백엔드 또는 공개키 매핑을 변경 또는 업데이트 하거나, (b) 새 유저가 시스템에 등록될 때에만 관여합니다. 새 도메인/유저를 등록할 때 라우팅 파일 해시를 블록체인에서 발표해야 합니다. 블록체인은(아틀라스 네트워크와 비교할 때) 확장성의 병목 지점이 될 수 있지만, 모든 유저에게 매우 드문 상황입니다. 또한 오프체인(off-chain) 이름 등록부를 이용해 100명이 넘는 유저가 단일 블록체인 트랜잭션에 등록할 수 있으므로 하루에 수십만 명의 유저 등록을 지원할 수 있습니다(기존 클라우드 기반 플랫폼의 하루 신규 유저 수와 비슷한 수준). 실제로 수십억 명의 유저에게 가이아를 확장하면 현재 불분명한 확장성 문제를 해결할 것이며, 이 과제 해결을 위해 지속적으로 연구하고 향후 과제로 삼아야 합니다.

5. 검증

유저 계정은 인터넷 애플리케이션을 사용하는 데 필수적입니다. 블록스택은 비밀번호 없이 모든 앱에서 작동하는 범용 유저네임을 유저에게 제공합니다. 비밀번호 기반 인증 대신 유저는 공개키 암호화를 통해 인증합니다. 즉, 로컬로 운영되는 소프트웨어 클라이언트는 해당 앱의 로그인 요청을 처리하여 인증 요청에 서명합니다. 블록스택의 인증 프로토콜인 Blockstack Auth는 앱을 유저의 가이아 허브 및 앱 별 개인키와 연결합니다. 앱이 이 정보를 사용하여 유저와 유저의 데이터를 저장하고 다른 유저가 생성한 데이터의 진실 여부를(authentic) 검증합니다.

5.1 단일 접속(Single Sign-On)

Blockstack Auth는 인증을 위해 퍼블릭키 암호화를 사용합니다. 유저는 앱에 접속하여 앱이 서명된 데이터를 생성하고 저장할 수 있도록 하며, 다른 유저는 이를 읽고 검증할 수 있습니다. 결과적으로 이는 접속한 유저가 적법하다고 다른 유저에게 증명하게 됩니다.

블록스택에서 접속의 목적은 앱 클라이언트에게 검증된(authentic) 데이터를 생성하고 저장할 수 있도록 충분한 정보를 제공하기 위해서 입니다. 즉, Auth 기능이 검증 프로그램의 형태로서 유저의 컴퓨터에서만 운영될 수 있다는 의미입니다. 스택스 블록체인에 모든 이름이 등록되어 있고, 각각의 앱과 검증 앱이 (1) 존재하는 모든 이름, (2) 모든 공개키와 가이아 허브의 최신 버전을 가지고 있기 때문입니다. 따라서, 서버 측 ID 제공자의 필요성이 사라졌습니다.

앱 클라이언트는 유저 데이터를 검증하기 위해 스택스 블록체인 피어에게 연락만 취하면 됩니다. 이를 위해, 유저는 접속중인 자신이 선호하는 스택스 피어의 네트워크 주소를 앱에 제공합니다.

유저는 “로그인” 버튼을 누르고 블록스택 앱에 접속합니다. 앱은 접속 요청과 함께 유저를 (blockstack.js SDK를 통해) 블록스택 검증앱(authenticator)으로 리디렉팅합니다. 유저는 접속할 때 사용할 블록스택 ID 선택권과 앱이 유저로부터 받아야 할 허가 리스트를 제공받습니다. ID를 선택하면, 검증앱이 유저를 다시 앱으로 디렉팅하고, 앱에 세가지 정보

를 전달합니다.

1. 유저의 유저네임(유저네임이 없다면 유저 공개키의 해시).
2. 유저 데이터를 암호화하고 서명하기 위한 각 어플리케이션 별 개인키. 이는 유저의 마스터 개인키, 접속을 위해 사용했던 ID, 앱의 HTTP Origin 을 통해 결정 및 생성됩니다.
3. 다른 유저와 데이터를 검색하기 위한 유저의 가이아 허브 및 선호하는 스택스 블록체인 피어의 URL.

이 과정을 거치면 유저는 유저네임을 제시하고 유저 데이터의 저장 위치를 앱에 지시할 수 있다. 이 후로 앱은 앱별(application-specific) 데이터를 끊임없이 읽고 쓸 수 있고, 다른 유저의 앱 별 데이터에 접근 가능합니다. 자체 스토리지나 ID솔루션 없이 가능합니다.

접속은 단순히 앱의 로컬 상태를 삭제(clear)하는 것이 목적이고, 이 과정에서 웹브라우저와 클라이언트가 앱 별 개인키를 잊게 됩니다.

6. 블록스택 라이브러리& SDK

Blockstack PBC(Public Benefit Corp, 블록스택 프로토콜 및 SDK 개발 기업)은 오픈 소스 참여자와 함께 블록스택을 위한 코어 프로토콜과 개발자 라이브러리를 개발합니다. 개발자 라이브러리를 사용하면 개발자가 블록스택 네트워크에서 앱을 보다 쉽게 만들 수 있으며, 유저는 블록스택 클라이언트를 통해 블록스택 네트워크의 다양한 구성 요소 및 다양한 앱과 상호 작용할 수 있습니다.

6.1 개발자 라이브러리

블록스택은 개발자가 최대한 쉽게 탈중앙 앱을 개발할 수 있도록 설계되었습니다. 스택스 블록체인 또는 탈중앙 스토리지와 상호 작용의 복잡성이 거의 드러나지 않아, 앱 개발자는 앱 로직에만 집중할 수 있습니다. 블록스택 오픈소스 저장소에는 다양한 플랫폼의 개발자 라이브러리를 포함하고 있습니다. 예를 들면, Javascript Web SDK (blockstack.js), iOS 및 Android 용 모바일 SDK가 있습니다. 이 모든 라이브러리는 MIT 라이선스의 조항에 따라 제공되며 <https://github.com/blockstack>에서 관련 내용을 확인하세요.

개발자 라이브러리는 인증 프로토콜 실행, 가이아 서버와의 직접 상호작용 및 스택스 트랜잭션 생성에 필요한 모든 API와 코드를 제공합니다. 라이브러리를 사용하면 개발자가 전통적인 앱 개발 만큼 쉽게 유저의 보안 및 개인 정보를 존중하는 탈중앙 앱을 개발할 수 있습니다.

Radiks 복잡한 소셜 그래프에서 데이터를 공유하려는 앱의 경우, 데이터 위에 인덱스를 구축하는 것이 유용하고 효율적입니다. Radiks 시스템은 이러한 인덱스를 만들고 상호작용하기 위한 서버 및 클라이언트 라이브러리입니다. Radiks 라이브러리를 사용하면 개발자가 앱 내에서 유저간 교차 가능한 구조화된 데이터 컬렉션을 만들 수 있고, 이는 필드값에 따라 쿼리될 수 있습니다. 여기에는 인덱스와 쿼리를 처리하는 서버 측 컴포넌트가 필요하지만, 중요한 것은 이는 유저의 신뢰할 수 있는 컴퓨팅 베이스에 포함되지 않습니다. 인덱스 넘어 쿼리를 작성하고 응답하는 데 필요한 데이터 암호 텍스트 및 일부 메타 데이터 만 볼 수 있습니다. Radiks에 대한 자세한 내용은 [26]을 참조하세요.

6.2 유저 소프트웨어

앱 개발자가 SDK 및 라이브러리를 사용하여 블록스택 네트워크와 상호 작용하는 반면, 유저 또한 유저 이름 등록, 가이아 서버 지정 및 응용 프로그램 인증과 같은 기능을 수행하기 위한 소프트웨어가 필요합니다. 블록스택 생태계는 현재 유저가 네트워크와 상호 작용할 수 있는 두 가지 오픈소스 프로젝트를 제공합니다.

1. **블록스택 브라우저.** 현재 인증(authenticator) 앱의 참조 오픈 소스를 구현한 것이며, 유저는 사용 가능한 블록스택 앱을 탐색하여 유저네임 등록과 앱 인증을 할 수 있습니다. 개인 데스크탑에 로컬로 설치하거나 웹 배포 양식으로 설치할 수 있습니다.
2. **블록스택 CLI.** 파워 유저와 개발자가 블록스택의 프로토콜과 상호 작용할 수 있게하는 커맨드 라인 유틸리티입니다. 인증 기능을 제공할 뿐만 아니라, 유저가 원시 트랜잭션을 생성하고 가이아를 사용해 고급 데이터 관리 작업을 수행할 수 있도록 합니다.

6.3 문서(Documentation)와 커뮤니티 리소스

블록스택 오픈소스 커뮤니티는 튜토리얼, API문서, 시스템 설계 문서를 관리합니다. 관련 자료는 Github 과 <https://docs.blockstack.org>에서 확인하세요.

블록스택 유저와 개발자는 다음과 같은 공식 커뮤니티 리소스를 사용할 수 있습니다.

- **Github:** 모든 소프트웨어 개발은 Github (<https://github.com/blockstack>)에서 이루어집니다.

- **포럼:** 파워 유저와 개발자는 블록스택 포럼 (<https://forum.blockstack.org>)에서 기술 관련 질문 대답, 아이디어 공유하고 서로에게 도움을 제공합니다.
- **슬랙(Slack):** 블록스택 커뮤니티는 실시간 채팅을 위해 퍼블릭 슬랙 그룹 <https://blockstack.slack.com>을 이용합니다.

7. 앱과 서비스

2019년 초 기준, 100개 이상의 앱이 블록스택에 구축되었습니다. 개발자는 다양한 유형의 앱을 구축하고 있으며 더욱 더 증가하는 블록스택 앱의 전체 목록을 app.co에서 확인할 수 있습니다. 블록스택은 모듈식이므로 다른 앱은 서로 다른 컴포넌트를 독립적으로 사용할 수 있습니다. 몇몇 사용 예시를 아래에서 기술하겠습니다.

현재 블록스택의 사무용 생산성 앱[27, 28, 29, 30]은 블록스택 Auth 및 가이아 스토리지를 이용하여 유저가 문서를 작성, 편집 및 공유할 수 있도록 합니다. 사용자가 서로의 문서를 쉽게 찾을 수 있도록 블록스택 프로필 검색 인덱서(indexer)를 사용합니다. 검색 인덱서는 탈중앙화되어 있습니다. 즉, 일련의 프로필이 전 세계적으로 표시되고 검색 가능하기 때문에 누구나 프로필 검색 인덱서를 배포하고 실행할 수 있습니다.

블록스택 생태계에는 많은 소셜 앱이 있습니다[31, 32, 33, 34]. 일반적으로 블록스택 Auth를 Radiks 서버 배포와 함께 사용하여 유저가 다른 유저의 데이터를 효율적으로 검색하고 가져올 수 있도록 합니다. 1개 이상의 앱이 전용 릴레이 채널을 사용하여 암호화 된 메시지를 다수의 유저에게 라우팅합니다[31]. 데이터 발행(publishing) 및 스토리지 앱[35, 36, 37, 38, 39, 40]은 가이아를 사용해 유저 데이터를 저장할 뿐만 아니라 기존 HTTP URL을 통해 이 데이터를 비블록스택(non-Blockstack) 유저와 공유합니다.


개발자 보상 스택스 블록체인이 마이닝 개념을 확장하여 앱 개발자가 네트워크에 고품질 앱을 출시하여 스택스 토큰을 "마이닝"할 수 있도록 했습니다. '앱 마이닝'이라고 하는 이 메커니즘은 네트워크에서 고품질 앱을 얻기 위한 보상 메커니즘으로 설계되었습니다. 앱 마이닝 프로그램은 현재 Blockstack PBC가 여러 개별 리뷰어와 함께 운영합니다. 개발자는 한 달에 한 번 앱 리뷰 신청서를 제출할 수 있고, 앱 순위 메커니즘의 순위에 따라 보상금(pay-out)을 받을 수 있습니다. 일련의 독립 리뷰어가 앱을 검토하고, 이 리뷰어는 좋은 앱을 평가를 위한 자체 기준을 가지고 있습니다. 앱 평가 집계 점수를 통해 앱 순위가 결정되고, 순위와 보상금은 매달 발표됩니다. 앱 마이닝 프로그램에 대한 자세한 내용은 이 백서에 포함되지 않으므로, 자세한 내용은 <https://app.co/mining>을 참조하세요.

8. 결론

블록스택은 탈중앙 앱 개발자에게 풀스택을 제공하는 탈중앙 컴퓨팅 네트워크입니다. 현재까지 100 개가 넘는 탈중앙 응용프로그램이 네트워크 상에 구축되었습니다. 블록스택을 사용하면 개발자는 서버와 데이터베이스를 운영할 필요가 없고 앱이 대신해 데이터를 유저가 제어하는 개인 데이터 보관함(locker)에 씁니다. 탈중앙 스토리지 시스템은 기존 클라우드 스토리지에 필적하는 성능을 제공하며 암호화 및 해독을 위한 약간의 오버헤드만 도입합니다. 인증 프로토콜을 이용하면 암호 인증보다 안정성이 낮은 비밀번호 입력 기반 로그인의 필요성이 사라집니다. 유저는 모든 서비스와 앱 이용에 단일 계정을 사용하므로 서비스 마다 새로운 계정을 만들 필요가 없습니다. 개발자 라이브러리를 통해 전통적인 인터넷 앱을 구축하는 것처럼 쉽게 플랫폼에서 탈중앙 앱을 개발할 수 있습니다.

이 백서는 블록스택의 최신 설계를 제시하고 있습니다. 초기 블록스택의 2016년 제작 및 2017년 구원 이후 핵심 설계는 진화를 거듭해 왔고, 제작 및 구축과 탈중앙 앱 개발자의 피드백을 통해 많은 교훈을 얻었습니다. 초기 백서(2017)와 달라진 주요 변경 사항은 (a) 새로운 블록체인을 보안 부트스트랩(bootstrap)하기 위한 조정가능 증명(Tunable Proofs) 메커니즘을 사용하는 스택스 블록체인에 대한 설명 추가, (b) 스마트 컨트랙트의 보안과 예측가능성에 중점을 두는 새로운 스마트 컨트랙트 언어에 대한 설명 추가입니다. 블록스택은 오픈소스로 발표됐습니다 [41].

감사의 말

지난 수년간 많은 분들이 블록스택의 설계와 실현에 기여해 주셨습니다. 다양한 아이디어를 통해 프로젝트 초반에 상당히 기여한 Larry Salibra, Ken Liao, Guy Lepage, Patrick Stanley, John Light, 개발자 라이브러리와 SDK 개발에 도움을 준 Hank Stoeve, Shreyas Thiagaraj, Matthew Little, 제품 설계와 개발자 문서 작성에 도움을 준 Jeff Domke, Mark Hendrickson, Thomas Osmonson, Jasper Jansz, Mary Anthony에게 감사의 말을 전합니다. 특히, 인프라 개발에 힘써준 Jesse Wiley,  Tim Wells, 많은 조언과 피드백을 해준 Brittany Laughlin과 Diwaker Gupta에게도 다시 한번 감사드립니다. <https://github.com/blockstack>에서 블록스택에 도움을 주신 더 많은 분들의 성함을 확인할 수 있습니다. 수 많은 분들이 이 프로젝트에 도움을 주어 한 분씩 모두 다 거명하는 것이 불가능합니다. 블록스택 오픈소스 커뮤니티의 아낌 없는 지원에 다시 한번 감사드립니다.

참조

- [1]S. Sulyman, "Client-server model," *IOSR Journal of Computer Engineering*, vol. 16, pp. 57-71, 01 2014.
- [2]N. Perlroth, "Yahoo says hackers stole data on 500 million users in 2014," Sept. 2016. <http://nyti.ms/2oAqn0G>.
- [3]K. Granville, "Facebook and cambridge analytica: What you need to know as fallout widens," Mar. 2018. <https://nyti.ms/2HP4Dr3>.
- [4]R. Mcnamee, "I mentored mark zuckerberg. i loved facebook. but i can't stay silent about what's happening.," Jan. 2019. <http://time.com/5505441/mark-zuckerberg-mentor-facebook-downfall/>.
- [5]"Blockstack website," 2019. <http://blockstack.org>.
- [6]J. H. Saltzer, D. P. Reed, and D. D. Clark, "End-to-end arguments in system design," *ACM Trans. Comput. Syst.*, vol. 2, pp. 277-288, Nov. 1984.
- [7]D. D. Clark and M. S. Blumenthal, "The end-to-end argument and application design: The role of trust," *Federal Comm. Law Journal*, vol. 63, no. 2, 2011.
- [8]M. Ali, J. Nelson, R. Shea, and M. Freedman, "Blockstack: A global naming and storage system secured by blockchains," in *Proc. USENIX Annual Technical Conference (ATC '16)*, June 2016.
- [9]J. Nelson, M. Ali, R. Shea, and M. J. Freedman, "Extending existing blockchains with virtualchain," in *Workshop on Distributed Cryptocurrencies and Consensus Ledgers (DCCL'16)*, (Chicago, IL), June 2016.
- [10]M. Ali, J. Nelson, R. Shea, and M. J. Freedman, "Bootstrapping trust in distributed systems with blockchains," *USENIX ;login:*, vol. 41, no. 3, pp. 52-58, 2016.
- [11]Satoshi Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," tech report, 2009. [https:// bitcoin.org/bitcoin.pdf](https://bitcoin.org/bitcoin.pdf).
- [12]A. Narayanan, J. Bonneau, E. Felten, A. Miller, and S. Goldfeder, *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*. Princeton, NJ, USA: Princeton University Press, 2016.
- [13]V. Buterin, "A next-generation smart contract and decentralized application platform," tech. rep., 2017. <https://github.com/ethereum/wiki/wiki/White-Paper>.
- [14]"Filecoin: A Cryptocurrency Operated File Network," tech report, 2014. [http:// filecoin.io / filecoin.pdf](http://filecoin.io/filecoin.pdf).
- [15] <https://eos.io>.
- [16]T. Hanke, M. Movahedi, and D. William, "Dfinity technology overview series consensus system rev. 1," 2018. <https://dfinity.org>.
- [17]"Ethereum 2.0 specifications," 2019. <https://github.com/ethereum/eth2.0-specs>.
- [18]Eng Keong Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim, "A survey and comparison of peer-to-peer overlay network schemes," *IEEE Communications Surveys Tutorials*, vol. 7, pp. 72-93, Second 2005.
- [19]"Oauth." <https://oauth.net>.
- [20]"Blockstack Core: Stacks blockchain v1," 2018. <https://github.com/blockstack/blockstack-core/tree/v20.0.8.1>.
- [21]"SIP 001: Burn Election," 2019. <https://github.com/blockstack/blockstack-core/blob/develop/sip/sip-001-burn-election.md>.
- [22]I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," in *Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, SIGCOMM '01*, pp. 149-160, 2001.
- [23]"App mining." <https://app.co/mining/>.
- [24]"SIP 002: Smart Contract Language," 2019. <https://github.com/blockstack/blockstack-core/blob/develop/sip/sip-002-smart-contract-language.md>.
- [25]"Google Cloud Storage SLA." Retrieved from <https://cloud.google.com/storage/sla> May 2017.
- [26]"Radiks." <https://github.com/blockstack-radiks>.
- [27]J. E. Hunter, "Graphite docs," Feb. 2019. <https://app.graphite-docs.com>.
- [28]D. Travino, "Noteriot," Feb. 2019. <https://note.riot.ai/>.
- [29]"Forms.id," Feb. 2019. <https://forms.id>.
- [30]"Blockusign," Feb. 2019. <https://blockusign.io>.

- [31]P. Bhardwaj and A. Carreira, "Stealthy," Feb. 2019. <https://www.stealthy.im>.
- [32]A. Sewrathan, R. Adjei, and F. Madutsa, "Afari," Feb. 2019. <https://afari.io>.
- [33]T. Alves, "Recall," Feb. 2019. <https://app.recall.photos/>.
- [34]T. Alves, "Travelstack," Feb. 2019. <https://app.travelstack.club>.
- [35]J. E. Hunter, "Graphite publishing," Feb. 2019. <https://publishing.graphitedocs.com>.
- [36]"Decs," Feb. 2019. <https://app.decs.xyz>.
- [37]"Sigle," Feb. 2019. <https://app.sigle.io>.
- [38]"Xorbrowser," Feb. 2019. <https://xorbrowser.com>.
- [39]"Mevaul," Feb. 2019. <https://mevaul.com/>.
- [40]"Xordrive," Feb. 2019. <https://xordrive.io>.
- [41]"Blockstack source code release v20.0.8," 2019. <http://github.com/blockstack/blockstack-core>.