

니어프로토콜(NEAR Protocol)

해당 번역본은 이용자가 디지털자산 관련 정보를 편리하게 확인할 수 있도록 단순 참고용으로 번역된 것이며, 투자 권유를 목적으로 하지 않습니다.

그 내용이 정확하지 않을 수 있으므로 정확한 정보 습득을 위해서는 원문을 참고하시거나 원문 작성 주체 측에 문의하시기 바랍니다.

두나무는 해당 번역본 내용의 진실성 및 정확성을 보장하지 않고, 제공된 정보에 의한 투자 결과에 대하여 어떠한 책임도 지지 않으며, 투자 손실은 투자자에게 귀속됩니다.

NEAR 백서

Section 01

초록

NEAR는 시스템의 설계 방식, 애플리케이션의 구축 방식, 그리고 웹 자체의 작동 방식을 변화시킬 잠재력을 가진 탈중앙화 애플리케이션 플랫폼이다. 이는 복잡한 기술이지만 단순한 목표를 지닌다 — 개발자와 기업가가 돈과 신원과 같은 고가치 자산을 안전하게 보호하면서도, 소비자가 접근할 수 있을 만큼 성능적이고 사용하기 쉬운 애플리케이션을 쉽게 그리고 지속 가능하게 구축할 수 있도록 하는 것이다. 이를 위해 NEAR는 최종 사용자에게 직관적인 경험을 제공하고, 수백만 대의 장치에 걸쳐 용량을 확장하며, 개발자에게 지속 가능한 새로운 비즈니스 모델을 제공하기 위해 처음부터 설계되었다. 이를 통해 NEAR는 오픈 파이낸스(Open Finance)의 범위를 확장하고 오픈 웹의 미래를 구동할 만큼 강력한 유일한 커뮤니티 기반 클라우드를 창조하고 있다.

다음 섹션에서는 NEAR가 시스템의 핵심 기술을 설계하고 구현하는 접근 방식을 설명한다. 가능한 한 이해하기 쉬운 언어를 사용하며, 기술적 구현을 다루기 전에 기본 원리, 가치, 설계 의도를 중심으로 관련 내용을 설명할 것이다. 기술적 주제에 대한 추가적인 깊이 있는 내용은 관련 주제별 논문과 블로그 게시물에서 확인할 수 있다. 활발히 개발 중인 복잡한 시스템의 특성상, 본 문서와 그가 설명하는 기술의 내용은 언제든지 변경될 수 있음을 명시해야 한다. 사실, NEAR 접근 방식의 특징 중 하나는 빠르고 실용적인 반복(iteration)이다. 프로토콜에 대한 최신 정보는 <https://www.near.org/blog>의 블로그 게시물, <http://near.chat>의 실시간 채팅 채널, 그리고 <https://github.com/nearprotocol>의 참조 코드 저장소에서 확인할 수 있다.

Section 02

서론

오늘날 웹의 풍요로움은 “허가 없는 혁신(permissionless innovation)” — 즉, 누구의 허락도 받지 않고 콘텐츠와 애플리케이션을 창조할 수 있는 능력 — 을 활용한 수백만 명의 사람들의 결합된 노력에서 비롯되었다. 그러나 데이터에 대한 자유의 부족은 참여자들의 이익에 정면으로 반하는 환경을 초래하였다.

소수의 기업들은 네트워크 효과로 수많은 사용자를 끌어들이고, 그들의 데이터를 독점함으로써 사용자가 대안을 찾지 못하게 만들어 거대한 이용자 기반을 확보했다. 마찬가지로, 이들 거대 플랫폼은 애플리케이션들이 자신들의 생태계 위에 구축되도록 유도한 뒤, 그 애플리케이션들이 너무 성공적이 되면 접근을 차단하거나 이익에 반하는 방향으로 행동했다. 그 결과, 이러한 폐쇄적 생태계는 혁신을 억제하고 웹의 광대한 영역을 사실상 독점하게 되었다.

미래에는 새로운 기술을 활용해 과거의 허가 없는 혁신을 다시 가능하게 함으로써, 사용자에게 자유를 주고 애플리케이션이 그들의 이익에 반하지 않고 오히려 이를 지원하는 보다 개방된 웹을 만들 수 있다.

우리는 이미 금융 분야에서 이러한 자유의 힘을 목격했다. 비트코인과 같은 탈중앙화 디지털 통화 및 그 기반이 되는 블록체인 기술은 전통적인 은행 시스템 비용의 일부만으로 수십억 달러 규모의 개인 간(peer-to-peer) 거래를 가능하게 했다. 동일한 기반 기술은 또한 \$50억 달러 이상의 가상 자산 경제 참여자들이 이러한 자산을 추적하고, 소유권을 획득하며, 허가 없이 서로 간에 거래할 수 있도록 한다. 이는 실물 자산이 디지털 자산처럼 검증된 소유권과 추적성을 가지고 디지털 영역으로 진입할 수 있게 한다.

오늘날 우리가 본 것을 넘어, 데이터의 자유가 기본적으로 허가 없는 혁신을 가능하게 하는 웹은 새로운 형태의 소프트웨어 개발을 촉진할 것이다. 이 웹에서는 개발자들이 공개된 상태 구성요소(state components)로부터 신속히 애플리케이션을 구성하고, 사용자와의 기생적 관계에 의존하지 않고 소프트웨어 자체에서 가능해진 새로운 비즈니스 모델로 이를 구동할 수 있다. 이는 사용자와 더 정직하고 협력적인 관계를 가진 애플리케이션의 창조를 가속화할 뿐 아니라, 그 위에 구축된 전혀 새로운 비즈니스의 등장을 가능하게 한다.

이러한 새로운 애플리케이션과 그것을 구동하는 오픈 웹은 올바른 형태의 인프라에 의해서만 실현될 수 있다. 새로운 웹의 플랫폼은 단일 주체가 통제할 수 없어야 하며, 확장성 부족으로 사용이 제한되어서도 안 된다. 그것은 웹 자체처럼 탈중앙화된 설계를 가져야 하며, 저장된 가치가 사용자 허락 없이 검열, 수정, 삭제될 수 없도록 폭넓게 분산된 운영자 커뮤니티에 의해 지원되어야 한다. 또한, 그것은 안전해야 한다.

Section 03

왜 탈중앙화가 중요한가

NEAR는 미래의 오픈 웹을 실현하고 그 경제를 구동하기 위해 설계된 탈중앙화 애플리케이션 플랫폼이다. NEAR는 비트코인을 ‘죽지 않는 통화(unkillable currency)’로 만든 동일한 핵심 기반 기술을 사용하며, 여기에 커뮤니티 합의, 데이터베이스 샤딩, 그리고 사용성 측면의 최첨단 발전을 결합한다. 이 웹 위에서는 새로운 통화에서부터 새로운 애플리케이션, 그리고 새로운 산업에 이르기까지 모든 것이 창조될 수 있으며, 이는 완전히 새로운 미래의 문을 연다.

표면적으로는, 탈중앙화 블록체인 기반 플랫폼의 많은 설계 목표들이 기존 플랫폼을 이용하면 더 빠르고 더 저렴하게 달성될 수 있다.

예를 들어, 이더리움 블록체인에서 데이터를 저장하거나 연산을 수행하는 비용은 동일한 기능을 아마존 웹 서비스(AWS)에서 수행할 때보다 수천 배에서 수백만 배 이상 비싸다. 개발자는 언제나 탈중앙화 플랫폼보다 훨씬 적은 비용으로 “중앙화된” 애플리케이션 또는 심지어 중앙화된 통화를 구축할 수 있다. 이는 탈중앙화 플랫폼이 정의상 프로세스와 저장소에 다수의 중복성을 내포하기 때문이다.

그렇다면 왜 우리는 탈중앙화를 지원하기 위해 추가 비용을 지불해야 하는가?

그 이유는 모든 데이터가 동일하게 만들어지지 않았기 때문이다.

예를 들어 디지털 화폐의 소유권, 개인 신원, 자산의 권리(title)와 같은 가치의 특정 요소들은 매우 민감하다. 중앙화된 시스템에서는 다음과 같은 주체들이 자신이 다루는 모든 잔액의 가치를 직접 변경할 수 있다.

1. 애플리케이션 코드의 출시나 업데이트를 통제하는 개발자
2. 데이터가 저장되는 플랫폼
3. 애플리케이션 코드를 실행하는 서버

이들 중 누구도 악의적으로 행동할 의도가 없더라도, 정부, 경찰, 해커의 개입으로 인해 이들의 행위가 사용자를 해치는 방향으로 쉽게 전환될 수 있으며, 결과적으로 보호해야 할 잔액이 검열, 변조, 탈취될 수 있다.

일반적인 사용자는 잠재적 취약성이 있더라도 일상적인 데이터와 연산을 중앙화된 애플리케이션에 맡긴다. 일반적으로 오직 은행과 정부만이 부와 신원이라는 가장 민감한 정보를 안전하게 보관할 수 있는 신뢰를 얻는다. 그러나 이러한 기관들 또한 인간적 오만, 부패, 도둑질의 영향으로부터 자유롭지 않다.

예를 들어 2008년의 글로벌 금융 위기는 과도하게 레버리지된 은행 시스템을 신뢰하는 데 내재된 근본적인 문제를 드러냈다. 또한 전 세계의 정부가 위기 시기에 시민들에게 강력한 자본 통제를 시행하는 명확한 사례를 제공했다. 이와 별개로, 오늘날 대부분 혹은 전부의 민감한 데이터가 해커들에 의해 이미 유출되었다는 것은 상식에 가까워졌다.

반면, 완전히 탈중앙화된 시스템에는 “전원 끄기” 스위치가 없으며, 악의적인 세력이 그 위에 구축된 애플리케이션에 자신의 의지를 강제로 관철시킬 방법도 없다. 이를 달성하기 위해서는 연산과 데이터 저장 모두에서 상당한 중복성이 필요하다. 이 영역의 어떤 단일 실패 지점도 공격에 악용될 수 있기 때문이다. 저장되는 정보가 민감할수록 더 많은 중복성과 보안이 요구되며... 바로 그만큼 탈중앙화의 중요성도 커진다.

블록체인 기반 시스템은 이러한 탈중앙화를 위한 기반(substrate)이다. 블록체인의 불변성은 이러한 시스템을 구성하고 그 중복성을 구동하는 수많은 참여자들 간의 협력과 조정을 유도하는 데 필요한 기본 요소(예: 토큰)를 제공한다. 이러한 시스템이 한 번 가동되면, 사실상 “죽지 않는(unkillable)” 존재가 된다.

이러한 시스템 위에 애플리케이션을 구축하는 이점은 상당하다. 고도로 민감한 정보가 전 세계적으로 안전하게 이용 가능할 뿐 아니라, 통화(currency)가 매체의 기본 구성 요소로 존재한다. 이러한 탈중앙화 애플리케이션은 오늘날의 웹보다 더 복잡한 인프라 위에서 작동하지만, 개별 기업의 데이터 사일로에 갇혀 있는 현재의 웹이 제공할 수 없는 즉각적이고 전 세계적인 통화, 가치, 정보의 풀(pool)에 접근할 수 있다. 또한, 이 앱들이 보호하는 데이터는 애플리케이션이 아니라 최종 사용자 자신이 완전히 소유하고 통제한다는 점이 매

우 중요하다. 이는 탈중앙화 인프라 없이는 존재할 수 없었던 수많은 새로운 사용 사례를 가능하게 한다.

탈중앙화가 결정적으로 중요하지만, 모든 블록체인 기반 시스템이 실제로 탈중앙화되어 있는 것은 아니다. 탈중앙화는 여러 차원에서 측정될 수 있는 스펙트럼이며, 근본적으로는 시스템 자체를 파괴하기 위해 얼마나 많은 참여자를 부패시켜야 하는지, 그리고 그 가능성이 얼마나 되는지로 귀결된다. 시스템이 보호해야 할 자산이 중요할수록, 단지 구호로서의 탈중앙화가 아니라 진정한 탈중앙화를 달성하는 것이 더 중요하다. 이후의 섹션에서는 NEAR가 이러한 탈중앙화를 달성하기 위한 기술적 아키텍처를 설명할 것이다.

NEAR의 간략한 요약

NEAR는 NEAR 프로토콜 블록체인 위에서 구동되는 탈중앙화 애플리케이션 플랫폼이다. 이 블록체인은 전 세계 수백 대의 기계에서 작동하며, 새로운 웹을 위한 강력하고 탈중앙화된 데이터 레이어를 형성할 수 있을 만큼 허가 불필요(permissionless), 고성능(performance), 그리고 안전하게 조직되어 있다.

본질적으로 NEAR는 사용자에게 소유된 공유되고 안전한 자금, 신원, 데이터 풀에 접근할 수 있는 애플리케이션을 실행하기 위한 플랫폼이다. 기술적으로 말하면, NEAR는 파티션 내성 네트워킹(partition-resistant networking), 서버리스 컴퓨트(serverless compute), 분산 저장(distributed storage)의 특성을 결합하여 새로운 형태의 플랫폼을 만든다.

비교하자면, [아마존 웹 서비스\(Amazon's Web Services\)](#)와 [마이크로소프트 애저\(Microsoft's Azure\)](#)는 오늘날 웹 인프라의 상당 부분을 운영하며, 애플리케이션이 배포되는 가장 일반적인 “클라우드” 중 두 가지이다. 이러한 컴퓨팅 및 저장 클라우드를 구성하는 개별 서버들은 모두 단일 주체에 의해 통제된다. 이는 그 위에서 실행되거나 그 안에 저장되는 모든 것이 해당 기업 또는 그 기업에 강제력을 행사할 수 있는 정부 기관의 의지에 완전히 종속된다는 뜻이다. 데이터는 쉽게 손실되거나, 검열되거나, 변조되거나, 판매되거나, 해킹될 수 있다. 이는 오직 하나의 실패 지점(single point of failure)만 존재하기 때문이다.

이러한 클라우드 서버에 배포된 애플리케이션은 원래의 개발자나 그 자격 증명을 보유한 사람에 의해 지속적으로 수정될 수 있다. 이는 개발자에게는 소프트웨어 업데이트를 용이하게 하지만, 동시에 애플리케이션이 접근할 수 있는 모든 데이터가 동일한 개발자들에 의해 — 그들이 직접 수행하거나, 해킹당했거나, 정부 당국에 의해 강제되었든 — 검열, 변조, 탈취될 수 있음을 의미한다. 사용자의 데이터가 일반적으로 대규모 통합 데이터베이스에 저장되기 때문에, 이러한 개발자들은 바로 그 공격의 매력적인 표적이 된다.

결국, 개발자와 플랫폼 자체의 취약성이 결합되어, 이들 플랫폼에 저장된 모든 민감한 데이터는 본질적으로 위험에 노출된다.

반면, 이러한 애플리케이션을 호스팅하는 클라우드가 누구나 참여할 수 있는 전 세계적 커뮤니티에 의해 운영된다면, 그 위에 저장된 프로그램과 자산은 투명해지고 사실상 “죽지 않는(unkillable)” 존재가 된다. 이를 통해 사용자는 돈, 신원, 디지털 자산과 같은 의미 있는 것들을 안전하게 저장하고, 제3자의 허락이나 플랫폼 없이도 누구와도 안전하게 거래할 수 있다. 전 세계적으로 다수의 중복성이 존재하기 때문에 단일 실패 지점이 없으며, 클라우드 네트워크를 구성하는 커뮤니티 구성원 간의 합의(consensus)가 프로그램적으로 달성되기 때문에 보안이 보장된다.

개발자의 취약성을 제거하기 위해, 이 클라우드에 배포된 애플리케이션은 프로그램적으로 잠금(lock)되어 이후의 업데이트가 접근하는 상태(state)를 수정할 수 없게 할 수 있다. 본질적으로, 애플리케이션이 이러한 상태에 도달하면 자율적으로 작동하게 되며, 외부 간섭이나 오류 없이 기능을 지속적으로 수행할 수 있다고 신뢰할 수 있다. 이는 돈, 신원, 핵심 데이터 조각과 같은 고가치 자산을 안전하게 저장할 수 있게 한다.

비트코인은 이러한 전 세계 커뮤니티 운영 클라우드의 첫 번째이자 가장 기본적인 버전으로 볼 수 있으며, 주로 비트코인 디지털 통화를 저장하고 이동시키는 용도로 사용된다.

이더리움은 두 번째이며 약간 더 정교한 버전으로, 비트코인의 기본 원리를 확장하여 보다 일반적인 컴퓨팅 및 저장 플랫폼을 창조했지만, 아직 의미 있는 대중적 채택에는 이르지 못한 초기 기술(raw technology)이다.

NEAR는 그 이전의 것들을 넘어서는 진화를 대표하며, 대중적 채택을 위해 해결해야 할 세 가지 핵심 과제 — 사용성(usability), 확장성(scalability), 보안성(security) — 을 모두 해결한 최초의 탈중앙화 애플리케이션 플랫폼이다.

커뮤니티 클라우드 구축의 도전 과제

이와 같은 커뮤니티 기반 시스템은 단일 주체나 알려진 단체의 그룹에 의해 운영되는 중앙화된 “클라우드” 인프라와는 매우 다른 도전 과제들을 가진다. 예를 들어 다음과 같다.

1. 모든 사람에게 포용적이어야 하면서도 조작이나 장악으로부터 안전해야 한다.
2. 참여자는 자신의 기여에 대해 공정하게 보상받아야 하지만, 동시에 부주의하거나 악의적인 행동을 유발하는 잘못된 인센티브를 피해야 한다.
3. 선한 행위자가 올바른 균형점을 찾을 수 있도록 게임 이론적으로 안전해야 하며, 악의적인 행위자가 시스템에 부정적인 영향을 미치는 것을 적극적으로 방지할 수 있어야 한다.

이러한 인센티브를 완벽히 정렬시키는 유일한 방법은 자유시장 기반 시스템이며, 따라서 NEAR 플랫폼은 이를 모두 결합하는 매개체로 “NEAR”라는 토큰을 사용한다. 이 토큰은 전 세계 어디에 있든 클라우드 자원을 사용하는 이용자가 서비스 제공자에게 공정하게 보상할 수 있도록 하며, 동시에 이들이 선의로 운영되도록 보장한다.

탈중앙화를 유지하기 위해, 이러한 커뮤니티 기반 시스템은 허가 불필요(permissionless)해야 하며, 즉 누구나 참여할 기회를 가져야 한다. 이를 보장하기 위해 익명성은 매우 중요하며, 참여자의 신원을 공개할 필요는 없다. 이는 탈중앙화를 가능하게 하지만, 동시에 다양한 형태의 부정 행위를 초래할 수 있으므로, 시스템의 모든 메커니즘은 단일 개인이 하나의 계정뿐 아니라 백만 개의 계정을 동시에 제어할 수도 있다는 가정을 전제로 설계되어야 한다. 따라서 우리는 시스템의 참여와 거버넌스에 있어 “1 토큰 = 1 표(one token equals one vote)”의 원칙에 따라 운영한다.

이러한 시스템은 또한 견고한 탈중앙화의 필요성과 기술 및 커뮤니티가 발전하지 못하면 도태될 수 있다는 현실 사이에서 균형을 이루어야 한다. 따라서 커뮤니티의 장기적인 건강성을 유지하기 위해서는 플랫폼 자체 내에서 높은 수준의 탈중앙화와 강력한 보안 보장을 유지하는 동시에, 시간이 지남에 따라 기술을 효율적으로 진화시킬 수 있는 프로세스를 구축해야 한다.

Section 03

왜 NEAR인가?

오늘날의 블록체인은 상당한 진보를 이루었다. 2008년에 출범한 최초의 블록체인 비트코인은 네트워크 가치가 3,000억 달러를 넘는 가치 저장 수단(store of value)로 자리잡았으며, 2014년에 출범한 최초의 “글로벌 컴퓨터” 이더리움은 게임부터 탈중앙화 금융(DeFi)에 이르는 수천 개의 혁신적인 애플리케이션을 보유하고 있다.

그러나 안타깝게도 이러한 초기 네트워크와 그 이후 등장한 어떤 네트워크도, 그 위에 구축된 애플리케이션들이 대중적으로 채택되는 데 필요한 격차를 메우지 못했으며, 전체 웹을 지원할 수 있을 만큼의 확장성을 제공하지도 못했다.

이는 두 가지 핵심 요인에 기인한다.

1. 시스템 설계
2. 조직 설계

시스템 설계는 중요하다. 다른 플랫폼의 기술적 아키텍처는 사용성과 확장성 모두에서 심각한 문제를 야기하여, 극히 일부의 기술적 혁신가를 제외하면 채택이 거의 불가능하게 만들었다. 최종 사용자들은 애플리케이션 사용 중 97~99%의 이탈률을 경험하며, 개발자들은 애플리케이션을 만들고 유지하는 과정에서 끝없는 좌절을 겪는다.

이러한 문제를 해결하려면 현재의 프로토콜 아키텍처에 대해 근본적이고 복잡한 변경이 필요하다. 그러나 기존 조직들은 이를 구현할 능력을 입증하지 못했다. 대신, 그들은 명세 설계와 구현의 다년간의 지연을 반복하며, 그 결과 기술은 점점 더 뒤처지고 있다.

NEAR의 플랫폼과 조직은 위에서 언급한 문제들을 해결하기 위해 특별히 설계되었다. 기술 설계는 세계에서 가장 사용하기 쉽고 확장 가능한 탈중앙화 플랫폼을 만드는 데 집착적으로 집중되어 있으며, 이를 통해 전 세계 규모의 애플리케이션이 실제로 채택될 수 있도록 한다. 조직과 거버넌스 구조는 프로토콜을 빠르게 배포하고 지속적으로 진화시킬 수 있도록 설계되어, 결코 시대에 뒤떨어지지 않게 한다.

다음 섹션에서는 이러한 문제를 해결하는 핵심 기능들을 살펴본다.

주요 기능

현재 플랫폼, 그 개발자, 그리고 최종 사용자가 직면한 주요 문제들은 아래와 같이 해결된다. 이러한 기능의 구체적인 구현에 대한 자세한 내용은 본 문서의 다음 섹션에서 다룬다.

사용성 우선(Usability First)

새로운 블록체인 기반 플랫폼들은 일반적으로 기존 플랫폼에 비해 확장성을 제공한다는 점에서 자신들의 차별성을 주장한다. 그러나 플랫폼이 충분히 채택되어 그 처리량(throughput)이 실제로 필요하지 않다면, 확장성은 아무 의미가 없다. 비유하자면, 아무도 오지 않는 사막 한가운데에 10만 명을 수용할 수 있는 거대한 경기장을 짓는 것과 같다.

따라서 더 시급히 해결해야 할 중요한 문제는, 개발자들이 사용자가 *실제로 사용할 수 있는 유용한* 애플리케이션을 *쉽게* 만들 수 있도록 하고, 그러한 애플리케이션이 개발자에게 *지속 가능한 가치*를 창출할 수 있게 하는 것이다.

이러한 차원의 일부 개선은 기술 스택의 2층(레이어 2)에서 처리할 수 있지만, 가장 중요한 부분은 프로토콜 수준에서 수행되어야 하며 사후에 덧붙일 수 없다.

최종 사용자 사용성(End-User Usability)

개발자는 최종 사용자가 실제로 사용할 수 있는 애플리케이션만 구축한다. NEAR의 “점진적 보안 (progressive security)” 모델은 개발자가 사용자 온보딩을 지연시키고, 사용자가 “블록체인” 개념을 학습할 필요를 없애며, 애플리케이션을 사용하는 데 필요한 권한 요청 상호작용의 수를 제한함으로써, 사용자 경험을 기존의 웹 환경과 더 유사하게 만들 수 있도록 한다.

1. **간단한 온보딩(Simple Onboarding)**: NEAR는 개발자가 사용자를 대신하여 특정 행동을 수행할 수 있도록 하여, 사용자가 애플리케이션에 처음 접속할 때 지갑을 제공하거나 토큰과 즉시 상호작용하지 않아도 온보딩할 수 있게 한다. 사람이 읽을 수 있는 이름(`foobar.near`)을 가진 계정은 애플리케이션별 키를 추적하므로, 사용자 계정은 기존 웹에서 익숙한 “단일 로그인(Single Sign On, SSO)” 기능(예: “페이스북/구글/깃허브 등으로 로그인”)과 유사하게 활용될 수 있다.
2. **손쉬운 구독(Easy Subscriptions)**: 스마트 계약 기반 계정(contract-based accounts)은 특정 애플리케이션에 맞춘 구독 생성과 맞춤형 권한 부여(custom permissioning)를 용이하게 한다.
3. **익숙한 사용 방식(Familiar Usage Styles)**: NEAR의 경제 모델은 개발자가 인프라 비용을 숨기기 위해 사용자의 대신 비용을 지불할 수 있도록 하여, 기존 웹 사용 패러다임과 일치하는 사용자 경험을 제공한다.
4. **예측 가능한 가격(Predictable Pricing)**: NEAR는 플랫폼의 거래를 단순한 방식으로 가격화하여, 최종 사용자가 플랫폼 이용 시 예측 가능한 가격과 낮은 인지 부담(cognitive load)을 경험하도록 한다.

개발자 사용성(Developer Usability)

플랫폼의 개발자를 지원하기 위한 여러 핵심적인 사용성 개선 사항들이 포함되어 있으며, 이를 통해 개발자는 다른 어떤 플랫폼보다도 더 쉽게 학습하고, 개발하며, 테스트하고, 애플리케이션을 배포할 수 있다.

1. **익숙한 언어(Familiar Languages)**: NEAR 노드는 WebAssembly(WASM)를 실행하며, 이는 여러 인기 있는 언어에서 컴파일할 수 있다. 초기에는 스마트 계약을 보안성이 높고 포괄적인 프로그래밍 언어인 Rust로 작성할 수 있으며, Rust는 빠르게 인기를 얻고 있다. 또한 NEAR는 TypeScript와 매우 유사한 AssemblyScript로 작성된 계약도 지원한다. TypeScript는 Microsoft가 개발한 자바스크립트 기반 언어로, 타입 시스템을 갖추고 있으며 개발자들 사이에서 매우 널리 사용되고 있다. 향후에는 더 일반적인 프로그래밍 언어들이 지원될 예정이므로, 개발자가 플랫폼 위에서 애플리케이션을 구축하기 위해 완전히 새로운 언어를 배워야 할 필요가 없게 된다.
2. **강력한 개발 도구(Robust Tooling)**: NEAR의 개발 도구 모음(Development Suite)은 통합된 도구 세트를 통해 개발자의 워크플로우를 지원하도록 설계되어, 애플리케이션을 쉽게 빌드, 테스트, 배포할 수 있다. 플랫폼 상의 도구와 제공되는 API는 개발자에게 기존 웹 애플리케이션 개발에서 익숙한 경험을 제공한다. 여기에는 원클릭 배포(one-click deploy), 통합 단위 테스트(integrated unit testing), 간편한 프론트엔드 통합, 웹 브라우저의 개발자 콘솔을 통한 디버깅

등이 포함된다.

3. **개발자 비즈니스 모델(Developer Business Models)**: NEAR 프로토콜은 개발자가 생태계를 위해 만든 오픈 구성 요소의 사용량에 따라 리베이트(rebate) 형태로 보상을 제공함으로써, 이들이 만든 구성 요소를 수익화할 수 있도록 지원한다. 이에 대한 세부 내용은 후속 섹션에서 다룬다.
4. **예측 가능한 가격(Predictable Pricing)**: NEAR는 플랫폼의 거래를 단순한 방식으로 가격화하여, 개발자가 플랫폼을 사용할 때 예측 가능한 비용과 낮은 인지 부담을 경험할 수 있도록 한다.

확장성 두 번째(Scalability Second)

미래 지향적인 프로토콜은 확장을 위해 상태(state)와 처리(processing) 모두를 샤딩(sharding)해야 한다. 플랫폼이 광범위하게 채택되면, 단일 기계는 체인의 모든 정보를 저장하거나 모든 거래를 검증할 수 없게 된다.

NEAR는 추가 노드가 참여함에 따라 네트워크의 용량을 확장할 수 있도록 하는 샤딩 접근법을 사용한다. 이는 사용량이 충분히 높아질 때 네트워크 노드를 여러 샤드로 동적으로 분할하고, 그 샤드들 간에 연산을 병렬화(parallelize)함으로써 이루어진다. 이 접근 방식을 통해 네트워크는 수요가 증가함에 따라 지속적으로 확장할 수 있다.

최근 블록체인 커뮤니티의 많은 샤딩 연구는 거래를 샤드 내(intra-shard) 거래와 샤드 간(cross-shard) 거래로 구분하여, 전자에 최적화하고 후자에는 훨씬 느린 솔루션을 제공하는 경향이 있다.

NEAR 프로토콜은 거래가 기본적으로 여러 샤드를 동시에 다룰 것이라고 가정하며, 이는 임의의 스마트 계약에 매우 일반적인 동작이기 때문에, 이에 맞게 성능을 최적화한다.

효율적인 개발과 진화(Efficient Development and Evolution)

기존 플랫폼과 신규 플랫폼 모두가 직면한 주요 과제 중 하나는 플랫폼의 개발과 진화를 어떻게 관리하느냐이다. 플랫폼 자체는 탈중앙화되어야 하지만, 이를 업데이트하고 발전시키는 방법에는 여러 접근 방식이 존재한다.

기존 프로토콜들은 혁신의 속도를 따라가기엔 반복(iteration) 능력이 부족하다는 것이 증명되었다. 성공적인 차세대 프로토콜은 네트워크의 탈중앙화를 유지하면서도, 다음 기술 물결에 의해 방해받지 않도록 효율적인 개발 프로세스를 보장해야 한다.

NEAR의 초기 개발은 세계 최고의 엔지니어, 기업가, 기술 전문가로 구성된 팀에 의해 수행되고 있으며, 그 거버넌스 구조는 프로토콜이 지속적인 커뮤니티의 감시를 받으면서도 충분한 효율성을 유지하여 출시 이후에도 경쟁력 있고 관련성을 유지하도록 설계되었다.

이를 통해 NEAR는 산업의 절반을 괴롭혀온 “출시 실패(failure to launch)” 문제를 피할 뿐 아니라, 나머지 절반을 묶어둔 “개선 실패(failure to improve)” 문제도 피할 수 있다.

진정한 탈중앙화(Real Decentralization)

비트코인과 이더리움은 종종 높은 수준의 탈중앙화를 이룬 것으로 평가받지만, 실제로는 여러 중앙화 문제를 겪고 있다. 예를 들어 비트코인의 경우, 전체 채굴 파워의 53%가 단 3개의 풀에 의해 통제된다. 게다가 현재 채굴 노드를 운영하려면 고가의 하드웨어가 필요하며, 이는 진입 장벽을 높이고 시간이 지남에 따라 노드 참여 인센티브를 감소시킨다.

신규 네트워크들은 종종 탈중앙화의 이상을 포기하고, 제한된 검증자 집합이나 완전히 “허가형 네트워크(permissioned network)”를 사용하는 더 중앙화된 구현을 통해 운영 효율성을 얻는 방향을 택한다. 그러나 이는 진정한 탈중앙화 네트워크의 근본 원칙 중 하나 — 즉, 네트워크의 가치는 독립적인 노드들 간의 중복성 수준에 의해 보호된다는 원칙 — 을 위배하는 것이다.

진정한 탈중앙화를 유지하기 위해 네트워크는 잠재적 노드 운영자의 허가 불필요(permissionless)한 참여를 허용해야 하며, 풀링(pooling)을 장려해서는 안 된다. 이러한 문제를 해결하기 위해 NEAR는 “임계값 지분 증명(Thresholded Proof of Stake)”이라 불리는 스테이킹 메커니즘을 사용한다. 이는 결정론적이면서도 광범위하게 공정하도록 특별히 설계되어, 대형 검증자들의 풀링을 유도하지 않으며 폭넓은 노드 참여를 장려한다.

노드의 진입 장벽을 낮추는 것은 단순히 네트워크를 탈중앙화하는 것 이상의 의미를 가진다. NEAR와 같은 수평적 확장(horizontally scalable) 시스템에서는 참여할 수 있는 노드가 많을수록 네트워크의 확장성 또한 증가하기 때문이다.

개발자와 기업가를 위한 새로운 비즈니스 모델

이더리움과 같은 블록체인의 초기 활용 사례 중 하나는 프로젝트가 자체 토큰을 발행하고 “초기 코인 공개(Initial Coin Offering, ICO)”를 통해 자금을 조달할 수 있도록 하는 것이었다. 이는 오픈소스 소프트웨어 및 인프라 세계에서 오랫동안 결여되어 있던, 인프라 개발자가 자본에 접근하고 프로젝트의 네트워크 효과를 촉진할 수 있는 혁신적인 새로운 방식처럼 보였다.

그러나 애플리케이션 계층의 토큰을 생성하는 것은 사용자에게 주요한 사용성 장벽을 초래했으며, 이후 발생한 투기 및 사기 사례는 대부분의 개발자에게 이 방식이 실행 가능한 길이 아님을 명확히 했다. NEAR는 개발자와 기업가에게 그들의 인프라를 수익화할 수 있는 보다 견고하고, 덜 침해적이며, 더 합법적인 방법을 제공한다. 스마트 계약이 호출될 때, 네트워크에서 발생한 수수료의 일부가 자동으로 해당 계약에 할당되며, (그렇게 코드가 작성되어 있다면) 개발자가 이를 인출할 수 있다. 이는 초기 인프라 개발을 장려함과 동시에(초기 계약들이 네트워크 사용을 촉진하는 네트워크 효과를 구축하기 때문에) 애플리케이션 및 인프라 개발자가 별도의 불필요한 토큰을 발행하지 않고도 자신의 창작물로부터 이익을 얻을 수 있는 비즈니스 모델을 제공한다.

Section 04

설계 원칙(Design Principles)

NEAR 플랫폼의 설계와 개발은 몇 가지 핵심 원칙에 의해 이끌어진다. 이러한 원칙들은 오늘날의 중앙화 및 탈중앙화 시스템 모두에 내재된 문제들을 반영한다.

1. **사용성(Usability)**: 플랫폼에 배포된 애플리케이션은 최종 사용자에게 원활하게 사용 가능해야 하

며, 개발자에게는 원활하게 구축 가능해야 한다. 가능한 한 기본 기술은 배경으로 사라지거나 최종 사용자로부터 완전히 숨겨져야 한다. 또한 개발자는 가능한 한 익숙한 언어와 패턴을 사용하여 개발 과정을 진행할 수 있어야 한다. 기본적인 애플리케이션은 직관적이고 간단하게 만들 수 있어야 하며, 보다 복잡한 애플리케이션 또한 보안성을 유지해야 한다.

2. **확장성(Scalability)**: 플랫폼은 기업 수준의 전 세계적으로 사용되는 애플리케이션을 지원하기 위해, 경제적 타당성이 존재하는 한 상한선 없이 확장 가능해야 한다.
3. **단순성(Simplicity)**: 시스템의 각 구성 요소는 그들의 주된 목적을 달성하기 위해 가능한 한 단순하게 설계되어야 한다. 이론적 완벽함보다는 단순성, 실용성, 그리고 이해의 용이성을 우선시해야 한다.
4. **지속 가능한 탈중앙화(Sustainable Decentralization)**: 플랫폼은 단기적, 장기적으로 모두 상당한 수준의 탈중앙화를 장려해야 하며, 이를 통해 플랫폼이 보유한 가치를 적절히 보호해야 한다. 플랫폼과 커뮤니티는 폭넓고 허가 없이 포용적이어야 하며, 탈중앙화와 참여를 적극적으로 장려해야 한다. 지속 가능성을 유지하기 위해 기술적 및 커뮤니티 거버넌스 메커니즘은 실용적인 반복(iteration)을 허용하되, 장기적으로 어떤 단일 주체에 의한 장악(capture)을 방지해야 한다.

Section 05

NEAR의 작동 방식(How NEAR Works)

NEAR는 탈중앙화 애플리케이션을 배포하고 실행하기 위한 커뮤니티 운영형 클라우드 인프라를 제공한다. 이는 탈중앙화 데이터베이스의 기능과 서버리스 컴퓨팅 플랫폼의 기능을 결합한다. 이 플랫폼의 구동을 가능하게 하는 토큰은 또한 그 위에 구축된 애플리케이션들이 새로운 방식으로 상호 작용할 수 있도록 한다. 이러한 기능들이 결합되어, 개발자는 돈, 신원, 자산과 같은 고위험(high-stakes) 데이터를 다루는 애플리케이션을 위해 검열 저항적인 백엔드(back-end)를 구축하고, 서로 원활하게 상호작용하는 개방형 상태 구성 요소(open-state components)를 만들 수 있다.

이러한 애플리케이션 백엔드와 구성요소들은 “**스마트 계약(smart contracts)**”이라 불리지만, 여기에서는 이들을 간단히 “**애플리케이션(applications)**”이라고 지칭한다.

이 클라우드를 구성하는 인프라는 전 세계 개인 및 조직이 운영하는 잠재적으로 무한한 수의 “노드(nodes)”로 이루어져 있다. 이들은 자신의 CPU 및 하드디스크 공간 일부를 제공하며, 이는 개인용 노트북일 수도 있고, 더 일반적으로는 전문적으로 배포된 서버일 수도 있다. 개발자는 스마트 계약을 작성하여 이 클라우드에 배포하며, 이는 기존 중앙화 클라우드에 애플리케이션을 배포하는 것과 매우 유사한 방식으로 느껴진다. 개발자가 “스마트 계약”이라 불리는 애플리케이션을 배포하고 이를 변경 불가능(immutable)하도록 설정하면, NEAR 커뮤니티의 일부 구성원이라도 존재하는 한 그 애플리케이션은 계속 작동한다. 최종 사용자는 배포된 애플리케이션과 상호작용할 때, 오늘날의 수많은 웹 또는 모바일 애플리케이션을 사용할 때처럼 익숙한 인터페이스를 통해 이를 이용하게 된다.

아마존이나 구글이 호스팅하는 중앙화 클라우드에서는, 개발자가 매달 자신의 애플리케이션 사용량에 따라 비용을 지불한다. 예를 들어, 웹페이지 방문자가 생성한 요청 수에 따라 과금된다. NEAR 플랫폼에서도 유사하게, 사용자 또는 개발자가 이 인프라의 커뮤니티 운영자에게 사용량에 대한 보상을 제공해야 한다. 오늘날의 클라우드 인프라와 마찬가지로, NEAR는 시스템 혼잡과 같은 외부 요인에 크게 영향을 받지 않는, 이해하기 쉬운 지표를 기반으로 사용 요금을 산정한다. 이러한 요소들은 오늘날의 다른 블록체인 기반 시스템에서 개발자들에게 매우 복잡한 문제를 초래한다.

NEAR의 경제적 구조에 대한 자세한 내용은 Economics 섹션에서 다룬다.

중앙화된 클라우드에서는 결정이 해당 기업에 의해 일방적으로 이루어진다. 반면, NEAR의 커뮤니티 운영형 클라우드는 탈중앙화되어 있으므로, 모든 업데이트는 네트워크 참여자의 충분한 정족수(quorum)에 의해 최종적으로 승인되어야 한다. 미래에 대한 업데이트는 커뮤니티로부터 제안되고, 효율성과 보안의 균형을 유지하는 포용적인 거버넌스 프로세스를 거쳐 결정된다.

거버넌스 프로세스에 대한 자세한 내용은 Governance 섹션에서 다룬다.

익명의, 그리고 잠재적으로 악의적일 수도 있는 노드 운영자들이 올바르게 코드를 실행하도록 보장하기 위해, 그들은 “지분 증명(Proof of Stake)”이라 불리는 스테이킹 과정에 참여한다. 이 과정에서 그들은 일종의 보증금(deposit)으로 일정 가치를 자발적으로 예치하며, 부정한 행위가 입증될 경우 이를 몰수당하게 된다.

스테이킹 프로세스에 대한 자세한 내용은 Technology 섹션에서 다룬다.

NEAR 플랫폼의 구성 요소(Elements of the NEAR Platform)

NEAR 플랫폼은 여러 개의 개별 요소들로 구성되어 있다. 이 중 일부는 플랫폼 자체에 기본적으로 내장되어 있으며, 다른 일부는 플랫폼과 결합되거나 그 위에서 함께 사용된다.

NEAR 토큰(The NEAR Token)

NEAR 토큰은 NEAR 생태계의 근본적인 네이티브 자산이며, 모든 계정에서 그 기능이 활성화되어 있다. 각 토큰은 Ether와 유사한 고유한 디지털 자산으로, 다음과 같은 용도로 사용될 수 있다.

1. 거래 처리 및 데이터 저장에 대한 비용을 시스템에 지불하기 위해 사용된다.
2. 스테이킹 과정에 참여하여 네트워크의 검증 노드(validating node)로서 운영에 참여할 수 있다.
3. 거버넌스 프로세스에 참여하여 네트워크 자원의 배분 방식과 향후 기술적 방향을 결정하는 데 기여할 수 있다.

NEAR 토큰은 네트워크를 운영하는 모든 참여자의 경제적 협력을 가능하게 하며, 동시에 네트워크 위에 구축된 애플리케이션들 간의 새로운 행태를 가능하게 한다.

기타 디지털 자산(Other Digital Assets)

이 플랫폼은 다음을 포함하되 이에 국한되지 않는 고유한 디지털 자산을 쉽게 저장할 수 있도록 설계되어 있다.

- **다른 토큰(Other Tokens):** 다른 체인에서 브리지(bridge)를 통해 “래핑(wrapped)”된 토큰이거나, NEAR 플랫폼 위에서 새로 생성된 토큰은 기본 플랫폼을 이용해 손쉽게 저장하고 이동할 수 있다. 이를 통해 다양한 종류의 토큰이 플랫폼 위에서 상품과 서비스의 결제 수단으로 사용될 수 있다. “스테이블코인(Stablecoin)”은 특정 자산(예: 미 달러)의 가격에 연동되도록 설계된 특수한 토큰으로, 이와 같은 거래에 특히 유용하다.
- **고유 디지털 자산(Unique Digital Assets):** 토큰과 유사하게, 게임 내 수집품에서부터 실제 자산 소유권의 표현에 이르기까지 다양한 형태의 디지털 자산(“대체 불가능 토큰(Non-Fungible

Token, NFT)’’이라고도 불림)을 플랫폼을 통해 저장하고 이동할 수 있다.

NEAR 플랫폼(The NEAR Platform)

커뮤니티 운영형 노드 클라우드로 구성된 핵심 플랫폼은 NEAR가 제공하는 가장 기본적인 인프라이다. 개발자는 허가 없이(permissionlessly) 이 클라우드에 스마트 계약을 배포할 수 있으며, 사용자는 그 위에서 구동되는 애플리케이션을 자유롭게 사용할 수 있다. 애플리케이션은 소비자 대상 게임에서 디지털 화폐에 이르기까지 다양하며, 그 상태(state, 즉 데이터)를 플랫폼 위에 안전하게 저장할 수 있다. 이는 개념적으로 이더리움(Ethereum) 플랫폼과 유사하다.

플랫폼 상에서 연산, 네트워크 사용 또는 저장을 필요로 하는 모든 작업은 수수료(fee)의 형태로 플랫폼에 대한 지불이 요구되며, 이 수수료는 플랫폼의 검증 노드 커뮤니티에 분배된다. 이러한 작업에는 새로운 계정의 생성, 새로운 계약의 배포, 계약에 의한 코드 실행, 계약에 의한 데이터의 저장 또는 수정이 포함될 수 있다.

이러한 비용의 세부 사항은 Economics 섹션에서 다루며, 노드의 작동 방식에 대한 세부 내용은 Technology 섹션에서 설명한다.

플랫폼은 허가 없이(permissionlessly) 인터페이스할 수 있다. 프로토콜의 규칙만 준수한다면, 독립적인 개발자는 누구든 사전 허가 없이 트랜잭션 제출, 계정 생성, 또는 새로운 노드 클라이언트 실행과 같은 방식으로 플랫폼과 상호작용하는 소프트웨어를 작성할 수 있다.

NEAR 개발 도구 모음(The NEAR Development Suite)

NEAR 플랫폼은 독립적이고 허가 없이(permissionlessly) 사용하도록 설계되었지만, 개발자와 최종 사용자가 보다 쉽게 활용할 수 있도록 다양한 도구와 참조 구현(reference implementations)이 함께 제공된다. 이러한 도구에는 다음이 포함된다.

- **NEAR SDKs:** NEAR는 Rust와 AssemblyScript(타입이 추가된 JavaScript)를 지원하여 스마트 계약을 작성할 수 있도록 한다. 개발자에게 최적의 경험을 제공하기 위해, NEAR는 이 두 언어를 위한 표준 데이터 구조, 예제, 테스트 도구를 포함한 완전한 SDK를 제공한다.
- **Gitpod for NEAR:** NEAR는 [Gitpod](#)라는 기존 기술을 활용하여 개발자가 즉시(on zero time) 온보딩할 수 있는 환경을 제공한다. Gitpod는 온라인 통합 개발 환경(IDE)을 제공하며, NEAR는 이를 커스터마이징하여 개발자가 웹 브라우저에서 스마트 계약을 쉽게 작성, 테스트, 배포할 수 있도록 했다. [NEAR Examples](#) 웹사이트에는 신규 및 기존 개발자 모두가 원클릭으로 배포할 수 있는 템플릿이 포함되어 있어 NEAR에서의 빌드 과정을 최대한 간단하게 만든다.
- **NEAR Wallet:** 지갑(Wallet)은 개발자와 최종 사용자가 네트워크를 이용하기 위해 필요한 자산을 저장하는 기본 수단이다. NEAR Wallet은 참조 구현(reference implementation)으로서, 애플리케이션 개발자가 보다 효과적인 사용자 경험을 설계할 수 있도록 지원하는 점진적 보안 모델(progressive security model)과 원활히 작동하도록 설계되었다. 향후에는 네트워크 상의 스테이킹 및 거버넌스 프로세스에 사용자가 쉽게 참여할 수 있는 기능이 내장될 예정이다.
- **NEAR Explorer:** 스마트 계약의 디버깅 및 네트워크 성능 이해를 돕기 위해, Explorer는 블록체인에서 수집한 정보를 직관적인 웹 기반 형식으로 시각화한다.
- **NEAR Command Line Tools:** NEAR 팀은 개발자가 로컬 환경에서 손쉽게 애플리케이션을 생성, 테스트, 배포할 수 있도록 간단한 명령줄(Command Line) 도구 세트를 제공한다.

이 모든 도구는 커뮤니티에 의해 오픈소스 방식으로 개발되고 있으며, 누구나 자유롭게 수정하거나 배포할 수 있다.

Section 06

경제 구조(Economics)

NEAR 플랫폼을 구성하는 생태계는 근본적으로 경제적 동력에 의해 움직인다. 이 경제 시스템은 참여자들이 허가 없이도(platform permissionlessly) 플랫폼의 핵심 기능을 수행하기 위해 자발적으로 조직될 수 있도록 인센티브를 제공하며, 동시에 바람직하지 않거나 무책임하거나 악의적인 행동을 억제하기 위한 강력한 디스인센티브를 제공한다. 플랫폼이 효과적으로 작동하기 위해서는 이러한 인센티브가 단기적·장기적으로 모두 존재해야 한다.

근본적으로 NEAR 플랫폼은 자발적 참여자 간의 시장(marketplace)이다. 공급 측면에서는, 검증 노드(validator node) 운영자 및 기타 핵심 인프라 제공자들이 “커뮤니티 클라우드”를 구성하는 서비스를 제공할 수 있도록 충분한 보상이 주어져야 한다. 수요 측면에서는, 플랫폼을 사용하는 개발자와 최종 사용자가 간단하고 명확하며 일관된 방식으로 사용료를 지불할 수 있어야 하며, 이를 통해 사용 편의성을 높인다.

더 나아가, 경제적 메커니즘은 생태계 전체를 지원하는 데에도 활용될 수 있다. 미시적(micro) 수준에서는 가장 유용한 애플리케이션을 개발한 개발자에게 직접 보상함으로써 새로운 비즈니스 모델을 창출할 수 있으며, 거시적(macro) 수준에서는 교육에서 거버넌스에 이르는 다양한 분야에 참여하는 광범위한 생태계 구성원의 노력을 조율할 수 있다.

이러한 경제적 메커니즘 각각의 구체적인 적용 방식은 아래 섹션에서 설명된다. 다음에서는 NEAR의 핵심 설계 원칙이 경제 구조에 어떻게 반영되는지를 기준으로 삼아, 기존 접근 방식들과의 비교를 통해 그 구조를 살펴본다.

NEAR 경제 설계 원칙(NEAR Economy Design Principles)

NEAR의 전체 시스템 설계 원칙은 다음과 같은 해석에 따라 그 경제 설계에 반영된다.

1. **사용성(Usability):** 최종 사용자와 개발자는 네트워크 사용에 대해 예측 가능하고 일관된 가격 체계를 가져야 한다. 사용자는 결코 데이터를 영구적으로 잃어서는 안 된다.
2. **확장성(Scalability):** 플랫폼은 경제적으로 타당한 임계값(threshold)에서 확장되어야 한다.
3. **단순성(Simplicity):** 시스템의 각 구성 요소는 그들의 주된 목적을 달성하기 위해 가능한 한 단순하게 설계되어야 한다.
4. **지속 가능한 탈중앙화(Sustainable Decentralization):** 플랫폼에 검증 노드(validating node)로 참여하기 위한 장벽은 가능한 한 낮게 설정되어야 하며, 이를 통해 폭넓은 참여가 가능해야 한다. 시간이 지남에 따라 참여가 부와 통제력을 소수에게 집중시키지 않아야 한다. 먼 미래에 발생하는 개별 거래는 오늘날 이루어지는 거래만큼이나 안전해야 하며, 이는 그 거래가 수정하는 가치(value)를 보호하기 위함이다.

개요(Overview)

NEAR의 경제 구조는 네트워크 보안과 생태계 발전을 위한 적절한 인센티브를 유지하면서도, 개발자와 최종 사용자에게 가능한 한 간단하고 효율적인 경험을 제공하도록 최적화되어 있다.

다음은 시스템을 구동하는 핵심 개념들의 요약이다.

1. **임계값 지분 증명(Thresholded Proof of Stake):**

검증 노드 운영자는 네트워크에 희소하고 가치 있는 연산 자원을 제공한다. 그들이 수행하는 연산이 올바르게 작동하는 것을 보장하기 위해, 이들은 결과를 보증하는 형태로 NEAR 토큰을 “스테이킹(stake)”해야 한다. 만약 이들의 결과가 부정확한 것으로 판명되면, 스테이커(staker)는 자신의 토큰을 잃게 된다. 이는 네트워크를 보호하기 위한 근본적인 메커니즘이다. 시스템 참여 임계값은 알고리즘적으로 가능한 한 낮은 수준으로 설정되어, 각 “에포크(epoch)”(하루의 절반) 기간 내에 최대한 많은 검증 노드가 참여할 수 있도록 한다.

2. **에포크 보상(Epoch Rewards):**

노드 운영자는 서비스 제공에 대한 보상으로 전체 토큰 공급량의 약 4.5%에 해당하는 고정 비율의 “보안 수수료(security fee)”를 연 단위로 지급받는다. 이 비율은 스테이커의 참여 수준을 충분히 유지하여 네트워크 보안을 확보하는 동시에, NEAR 토큰의 다른 생태계 사용과의 균형을 맞추도록 설계되었다.

3. **프로토콜 재무금고(Protocol Treasury):**

검증자 외에도, 프로토콜 재무금고는 매년 전체 공급량의 0.5%를 수령하여 생태계 개발에 지속적으로 재투자한다.

4. **거래 비용(Transaction Costs):**

네트워크 사용은 두 가지 유형의 자원을 소비한다 — 즉각적 자원과 장기적 자원이다. 즉각적 비용은 모든 거래에서 발생하며, 이는 네트워크 자체와 일부 연산 자원의 사용으로 인해 발생한다. 이러한 비용은 대부분 예측 가능한 단위 거래당 비용 형태로 산정되며, NEAR 토큰으로 지불된다.

5. **저장 비용(Storage Costs):**

저장은 장기적 비용이다. 데이터를 저장하는 것은 네트워크 노드에 지속적인 부담을 주기 때문이다. 저장 비용은 계정이나 계약이 일정한 최소 잔액(minimum balance)의 NEAR 토큰을 유지함으로써 충당된다. 이는 검증자들이 자신의 노드에서 계약 및 계정 상태를 유지하는 대가로 인플레이션을 통한 간접적인 보상 메커니즘을 제공한다.

6. **인플레이션(Inflation):**

인플레이션은 검증자 및 프로토콜 재무금고로의 지급액에서 수집된 거래 수수료(및 이름 경매(name auction)와 같은 몇 가지 NEAR 소각 메커니즘)를 뺀 값으로 정의된다. 전체적으로 최대 인플레이션율은 5%이며, 네트워크 사용량이 증가하고 더 많은 거래 수수료가 소각됨에 따라 감소할 수 있다. 충분한 양의 수수료가 소각되는 경우, 인플레이션이 음수(즉, 총 공급량이 감소)로 전환될 가능성도 있다.

7. **확장 임계값(Scaling Thresholds):**

네트워크가 사용량에 따라 용량을 확장하는 구조를 갖는 경우, 추가 용량을 도입하는 기준이 되는 임계값은 경제적 요인에 의해 결정된다.

8. **보안 임계값(Security Thresholds):**

참여자 간의 바람직한 행위를 유도하는 일부 임계값은 경제적 인센티브를 통해 설정된다. 예를 들어 “피셔맨(Fishermen)” 메커니즘(별도로 설명됨)이 이에 해당한다.

이러한 원칙 각각의 정당성은 다음 섹션에서 보다 자세히 설명된다.

제공되는 자원(Resources Provided)

블록체인 기반 클라우드는 그 위에서 실행되는 애플리케이션에 여러 구체적인 자원을 제공한다.

- **연산(Compute, CPU):** 이는 계약 내 코드를 실행하는 실제 컴퓨터 처리 능력(및 즉시 사용 가능한 RAM)을 의미한다.
- **대역폭(Bandwidth, “Network”):** 이는 참여자와 사용자 간의 네트워크 트래픽으로, 트랜잭션을 제출하는 메시지와 블록을 전파하는 메시지를 포함한다.
- **저장(Storage):** 체인 상의 영구적 데이터 저장을 의미하며, 일반적으로 저장 공간(예: 킬로바이트) 단위로 표현된다.

이더리움(Ethereum)과 같은 기존 블록체인은 이러한 모든 자원을 단일 선불 트랜잭션 수수료에 포함시킨다. 이 수수료는 각각의 자원을 별도로 회계 처리하지만, 결국 개발자나 사용자는 이를 단 한 번에 지불한다. 이 수수료는 “가스(gas)” 단위로 표시되며, 매우 높은 변동성을 가진다.

개발자는 예측 가능한 가격을 선호하는데, 이는 예산을 계획하고 최종 사용자에게 명확한 가격을 제공하기 위함이다. NEAR에서 위 자원들의 가격은 완전한 경매 기반이 아니라, 시스템 사용량에 따라 서서히 조정되며(지속적으로 사용량이 증가할 경우 리샤딩(Resharding)의 완화 효과를 받는다), 이를 통해 개발자는 트랜잭션 실행이나 저장 유지 비용을 보다 안정적으로 예측할 수 있다.

초기에는 이러한 모든 자원이 NEAR 토큰으로 가격화되고 지불된다. 향후에는 스테이블코인과 같은 안정적인 통화 단위(예: USD에 연동된 토큰)로도 가격이 표시될 수 있다.

연산 및 대역폭(Compute and Bandwidth, “gas”)

연산(CPU)은 트랜잭션 실행 시 소모되는 일시적인 자원이다. 각 CPU 명령어의 비용은 “가스(gas)” 단위로 계산되며, 그 가격은 NEAR 토큰으로 표시되는 가스 가격의 점진적 조정에 따라 결정된다. 대역폭은 일반적으로 바이트 단위로 측정되지만, NEAR 플랫폼에서는 참조 하드웨어(reference hardware)를 기준으로 계산된 단순한 오버헤드 계수(coefficient of overhead)를 사용하여 이를 가스 단위로 변환한다.

각 청크(chunk, 즉 블록의 일부)는 참조 하드웨어에서 실행될 때 “하나의 블록이 담을 수 있는 최대량”을 기준으로 특정한 최대 가스 한도(max gas limit)를 가진다. 이 블록 한도는 성능 향상이나 더 좋은 하드웨어 사용에 따라 네트워크 참여자들이 시스템 거버넌스 절차를 통해 조정할 수 있다.

현재의 가스 가격은 **예측 가능하지만 고정되어 있지는 않다**. 각 블록마다 다음과 같은 방식으로 조정된다.

- 이전 블록이 절반 이상 채워졌다면, 이전 블록의 가스 가격은 α 라는 매개변수에 의해 증가한다.
- 이전 블록이 절반 미만으로 채워졌다면, 이전 블록의 가스 가격은 동일한 매개변수 α 에 의해 감소한다.

가스 사용량은 아래에서 설명하는 리샤딩(Resharding)을 유발할 수도 있다.

저장(Storage)

저장은 장기적으로 희소한 자산이다. 애플리케이션이나 사용자가 이를 사용하기 위해서는, 해당 계정이 사용하는 저장 용량에 비례하는 최소 잔액(minimum balance)을 유지해야 한다. 바이트당 요구되는 NEAR 토큰의 양은 고정되어 있으며, 주요 거버넌스 결정에 의해서만 변경될 수 있다. 또한 저장 하드웨어 및 시스템 용량의 발전 추세를 고려할 때, 향후 이 값은 하향 조정될 가능성이 높다.

예를 들어, 어떤 계약이 저장된 데이터로 인해 10킬로바이트의 저장 공간을 차지한다면, 해당 계약은 최소 1 NEAR의 잔액을 유지해야 한다. 이더리움의 USDT와 같은 계약은 사용 중인 저장 공간을 감당하기 위해 약 10,000 NEAR의 잔액을 유지해야 할 것이다. 이는 일반 사용자 계정 또한 일정 비율의 NEAR 최소 잔액을 유지해야 함을 의미한다. 계정에 NEAR 최소 잔액을 유지하는 것은, 이 금액이 스테이킹되거나 다른 애플리케이션에서 사용되지 않음을 의미한다. 검증자들은 인플레이션과 전체 스테이킹 총액이 더 작아지는 구조를 통해, 이 저장 상태를 유지하는 대가로 간접적으로 보상을 받는다.

NEAR 시스템은 각 샤드(shard)의 상태 크기를 대체로 균형 있게 유지하여, 각 노드가 거의 동일한 상태 용량(전체 상태 크기를 샤드 수로 나눈 값에 해당)을 관리하도록 한다. 개별 샤드의 상태 크기가 변화하면, 계정과 계약의 샤드 배정이 변경되어 이러한 균형이 유지된다.

리샤딩(Resharding)

각 계정(account)과 계약(contract)은 특정 샤드(shard)에 할당된다. 그러나 이러한 계약의 사용량은 균등하지 않기 때문에, 일부 샤드의 사용량이나 크기가 다른 샤드보다 훨씬 빠르게 증가할 수 있다. 이를 방지하기 위해 NEAR는 **리샤딩(resharding)** 메커니즘을 사용하여 특정 조건에 따라 주기적으로 샤드를 재균형화한다. 그 결과, 트랜잭션 및 저장 사용량이 합리적으로 균형 잡힌 샤드 집합이 형성된다.

각 에포크(epoch, 하루의 절반)마다, 해당 기간 동안 블록이 얼마나 가득 찼는지와 같은 통계 및 기타 관련 조건들이 축적된다. 각 계약은 이전 에포크 동안의 사용량과 현재 저장 상태를 기준으로 평가되며, 이후 계약들은 예상되는 총합 특성이 유사하도록 여러 “버킷(bucket)” 그룹으로 분류된다. 시스템은 샤드당 트랜잭션 사용량의 근사 한도(“가스 한도(gas limit)”)와 노드당 예상 저장량을 알고 있다. 만약 이전 에포크에서 사용된 자원이 특정 임계값(threshold)을 초과한 경우(예: 다수의 블록이 절반 이상 찼을 경우), 새로운 샤드가 추가로 할당되어 버킷의 수가 증가하고 각 샤드의 평균 사용량이 분산된다.

새로운 샤드가 추가되면, 검증자 좌석(seat)의 수도 늘어나게 되며, 좌석당 가격(NEAR 토큰 단위)이 낮아짐에 따라 더 많은 고유 검증자들이 네트워크에 참여할 수 있다. 이 계산은 실제로 새로운 샤드가 사용되기 **한 에포크 이전에 미리 수행**되어, 검증자들이 필요한 상태(state)와 기타 샤드 정보를 재동기화(re-sync)할 시간을 확보할 수 있도록 한다.

인플레이션(Inflation)

시스템의 전체 인플레이션(새로운 토큰 발행율)은 검증 노드를 운영하기 위한 에포크 보상의 규모에 따라 결정된다. 새로운 토큰 발행률은 연간 **최대 5%**로 상한(cap)이 설정되어 있다. 실제 인플레이션율은 각 에포크(1/2일) 단위로 계산되며, 이는 에포크별 예상 인플레이션율에서 해당 기간 동안 수집된 수수료를 차감한 값으로 산출된다. 플랫폼이 확보한 수수료는 인플레이션에서 차감되므로, 네트워크 사용량이 증가할수록 시스템의 전체 인플레이션은 감소한다. 만약 시스템의 사용 수수료가 인플레이션으로 생성되는 토큰의 양을 안정

적으로 초과한다면, 시스템은 전반적으로 **디플레이션(deflationary)** 상태가 된다.

시스템은 샤딩 구조를 가지며, 각 샤드의 검증자 배정이 시스템에 명시적으로 알려지지 않는 “비가시적(hidden) 검증자”를 포함하므로, 모든 보상을 **사회화(socialize)** 해야 한다. 즉, “보안 수수료(security fee)”는 각 샤드가 처리한 트랜잭션이나 저장량과 관계없이 모든 검증자에게 균등하게 분배된다.

경제적 이해관계자(Economic Stakeholders)

- **검증자(Validators):** 노드를 실행하여 네트워크의 연산 자원과 보안을 제공한다.
- **개발자(Developers):** 네트워크 위에서 실행되는 애플리케이션을 만든다.
- **토큰 보유자(Token Holders):** 토큰 잔액을 유지하는 계정 또는 애플리케이션이다.
- **NEAR 재단(NEAR Foundation):** 네트워크 참여자들의 거버넌스와 기술적 발전을 조정하는 독립 기관이다.
- **제3자 관찰자(Third Party Observers):** 체인을 감시하며 부정행위 및 악의적 행동으로부터 추가적인 보호를 제공한다.
- **사용자(Users):** 네트워크상의 애플리케이션을 사용하는 최종 사용자로, 토큰 잔액을 직접 유지하지 않을 수도 있다.

이러한 이해관계자 각각에게 경제 정책이 미치는 영향은 아래에서 상세히 살펴본다.

검증자 보상(Validator Rewards)

“지분 증명(Proof of Stake)” 시스템으로서 NEAR 플랫폼의 보안은, 노드를 운영하는 검증자(validators)가 자신의 토큰 일부를 예치물(deposit) 형태로 스테이킹(staking)하여 올바른 행동을 보증하고, 검증자 선출 시 시빌 공격(Sybil attack)을 방지하는 데서 비롯된다. 검증자가 잘못된 블록을 생성하거나 대체 체인(예: 이중 지출(double spend)을 유도하기 위한 목적)을 만들 경우, 해당 예치금은 **슬래싱(slashed)** 처리되어 삭감된다.

검증자는 “임계값 지분 증명(Thresholded Proof of Stake)” 모델을 기반으로 선택된다. 이 모델은 경매 방식을 사용하여 각 잠재적 검증자에게 몇 개의 “좌석(seats)”이 할당될지를 결정하며, 즉 하나의 좌석에 필요한 최소 토큰 수(스테이킹 임계값)를 계산한다. 이 경매는 공정한 기회(equality of opportunity)를 제공하도록 설계되어 있으며, 가능한 한 많은 참여자가 네트워크의 검증 과정에 참여할 수 있도록 하여 의미 있는 탈중앙화를 실현한다.

검증자는 자신이 네트워크 전체 스테이킹 비율에서 차지하는 비율에 따라 검증 과정에 참여할 수 있을 것으로 결정론적으로(expect deterministically) 예측할 수 있다. 각 검증자는 다음 세 가지 역할 중 하나를 맡을 수 있다.

1. 블록 생산자(Block Producer)
2. 청크 생산자(Chunk Producer)
3. 히든 검증자(Hidden Validator)

검증자에게 할당된 역할과 관계없이, 보상은 검증자가 전체 스테이크에서 차지하는 비율에 비례하여 분배된다. 따라서 검증자가 되기 위한 최소 스테이킹 한도에 도달하지 않아도 풀(pool)에 참여할 필요가 없다.

검증자는 블록 및 청크를 생성하고, 보안 및 데이터 가용성을 제공함으로써 네트워크에 기여한 대가로 매 에포크(epoch)마다 목표량의 NEAR 토큰을 보상받는다. 이 목표치는 연 기준으로 전체 공급량의 **4.5%**에 해당하도록 계산된다.

검증자는 에포크 단위로 선출되며, 각 검증자는 청크를 검증하고, 데이터 가용성을 제공하고, 블록을 생성하기 위해 동일한 양의 작업을 수행해야 한다. 따라서 보상은 각 에포크마다 분배되며, 검증자의 스테이크 비율에 따라 비례적으로 나누어진다.

각 에포크 동안 수집된 모든 거래 수수료(계약 리베이트로 할당되는 부분 제외)는 시스템에 의해 **소각(burn)** 된다. 인플레이션 기반 보상은 수수료의 양이나 소각된 금액에 관계없이 동일한 비율로 검증자에게 지급된다. 결과적으로, 시스템 전체의 인플레이션은 지불된 수수료의 양에 비례하여 감소하며, 네트워크 사용 수수료가 인플레이션 보상보다 많아질 경우 시스템은 **디플레이션(deflationary)** 상태가 된다.

검증 노드를 실행하기 위한 연산 요구 사항은 최소화되도록 설계되었다. 대부분의 운영자는 표준 클라우드 가상 머신(예: Amazon AWS의 월 \$100 인스턴스나 Google Cloud의 n1-highcpu-4 인스턴스)만으로도 손쉽게 운영할 수 있다. 다만, 다수의 샤드에서 동시에 참여할 만큼 많은 토큰을 스테이킹하려는 검증자는 추가 저장소, 대역폭, 연산 부하를 처리하기 위해 더 많은 하드웨어(및 중복성)를 사용하는 것이 바람직하다.

제3자 관찰자(“히든 검증자(Hidden Validators)”와 “피셔맨(Fisherman)”)

복잡하지 않은 샤딩 기반 블록체인 시스템의 주요 단점 중 하나는, 네트워크 전체 검증자 중 일부만 특정 샤드의 트랜잭션을 검증하기 때문에 전체 시스템의 보안이 분할된다는 점이다. NEAR는 이 문제를 해결하기 위해 두 가지 메커니즘 — “**히든 검증자(Hidden Validators)**”와 “**피셔맨(Fishermen)**” — 을 도입한다.

히든 검증자(Hidden Validators)는 일반 검증자 풀에서 선택된 후, 자신을 제외한 어떤 참여자에게도 알려지지 않은 샤드를 검증하도록 지정된다. 이 과정은 다른 섹션에서 자세히 설명되며, 이는 하나의 샤드 내에서 악의적 행위를 수행하기 위해 충분한 수의 노드를 부패시키는 것을 극도로 어렵게 만든다. 히든 검증자는 일반 검증자 보상 과정의 일환으로 청크 및 블록의 유효성을 검증하고 서명한 대가를 받는다.

피셔맨(Fishermen)은 무허가(permissionless) 방식으로 부정 행위를 탐지하고 보고하는 관찰 노드이다. 이 노드들은 네트워크와 동기화되어 있지만, 합의 과정에 직접 참여하지 않으며 특정한 활동에 대해 정기적인 보상을 받지 않는다. 피셔맨에는 지갑 운영자, 애플리케이션 개발자, 거래소 인프라 등이 포함될 수 있다. 이들은 자신에게 중요한 체인 부분을 검증하며, 문제가 발견되면 **챌린지(challenge)**를 통해 이를 신고할 수 있다. 챌린지의 악용(“그리핑(griefing)”)을 방지하기 위해, 사전에 **10 NEAR**의 소액 보증금이 예치되어야 한다.

시스템은 피셔맨 노드를 운영하는 데 대한 보상을 제공하지 않는다(즉, 성공적인 챌린지 제출에 대한 보상이 없다). 대신, 피셔맨 노드를 운영하는 참여자들은 네트워크의 보안을 유지하려는 **외부적 동기(external motivation)**를 가지고 있다.

계약 보상(Contract Rewards)

특정 트랜잭션에서 발생한 수수료의 일부는 해당 트랜잭션 중 실행된 **스마트 계약(contract)**에 다시 분배된다. 이러한 “**계약 보상(Contract Reward)**”은 계약 내에 명시된 규칙에 따라 배분될 수 있으며, 예를 들어 계약 개발자가 제어하는 계정, 투자자, DAO 등에 할당될 수 있다.

이 보상에 할당되는 수수료의 비율은 시스템 수준 매개변수로 설정되며, 초기 최소값은 **30%**이다. 개발자는 이 메커니즘 외에도 사용자가 호출 시 자금을 첨부하도록 요구함으로써 추가 요금을 부과할 수 있다.

이는 애플리케이션 사용에 대해 별도의 수익 모델을 마련하기 어려운 개발자에게 새로운 비즈니스 모델을 제공한다. 시스템 수준에서 최소 수수료를 설정함으로써, 경쟁으로 인해 보상이 0으로 수렴하거나 다른 개발자가 해당 수수료를 제거(forking out)하는 식의 “**하향 경쟁(race to the bottom)**”을 방지할 수 있다.

이 메커니즘은 개발자가 네트워크용 애플리케이션과 핵심 계약(core contracts)을 개발하도록 강력히 유도한다. 계약의 사용량에 비례하여 직접 보상을 받을 수 있기 때문이다.

토큰 보유자(Token Holders)

토큰 보유자는 스테이킹(staking) 과정에 참여하지 않기로 선택할 수 있다. 예를 들어, 일시적으로만 토큰을 보유하고 있거나, 거래 시장에 유동성을 제공하고 있거나, 단순히 참여를 원하지 않는 경우가 해당된다. 스테이킹 및 검증 과정에 참여하지 않는 토큰 보유자는 네트워크 운영으로부터 추가적인 혜택을 받지 않는다. 그러나 그들의 토큰은 여전히 데이터 저장 및 네트워크 상의 애플리케이션 사용을 가능하게 함으로써 **실질적 효용(utility)**을 가진다.

프로토콜 재무금고(Protocol Treasury)

프로토콜의 지속적인 커뮤니티 성장과 진화를 가능하게 하기 위해, 인플레이션의 일부(인플레이션의 10%, 즉 초기 파라미터 기준 연간 총 0.5%)가 **프로토콜 재무금고(Protocol Treasury)**에 할당된다. 향후에는 커뮤니티가 이 자금을 관리하여, **지속 가능한 생태계 발전 및 장기적 NEAR 생태계 개발(특히 자체적으로 지속되기 어려운 프로토콜 개발 등)**을 조정하는 역할을 맡게 될 것으로 예상된다.

현재 탈중앙화 거버넌스의 발전 수준을 고려할 때, 초기에는 재무금고가 **NEAR 재단(NEAR Foundation)**에 의해 감독된다. NEAR 재단의 사명은 전 세계 사람들에게 혜택을 주는 커뮤니티 주도 혁신을 가능하게 하는 것이다. 재단은 완전히 탈중앙화된 네트워크 운영에 필수적이지는 않지만, 다른 주체가 수행하기 어려운 방식으로 프로젝트의 지속적 발전을 지원할 수 있다. 예를 들어, 재단은 특정 비즈니스 모델이 없어 자금 지원을 받기 어려운 공익적 교육, 이벤트, 인프라 프로젝트 등을 지원한다.

프로토콜 재무금고에 할당된 인플레이션 금액은 **탈중앙화 경제적 힘(decentralizing economic force)**으로 작용한다. 이는 생태계 내 자본을 다시 분배하여, 스테이크를 보유하지 않더라도 공익을 지원하는 개발자 및 참여자들에게 보상을 제공할 수 있도록 하기 때문이다.

특별 조건(Special Conditions)

슬래싱 조건 및 점진적 슬래싱(Slashing Conditions and Progressive Slashing)

NEAR 플랫폼에서 발생할 수 있는 악의적 행위는 크게 두 가지 유형이 있다.

1. **이중 서명(Double Signing)**: 동일한 높이의 서로 다른 두 개 이상의 블록에 서명하는 행위.
2. **잘못된 청크(Invalid Chunks)**: 잘못된 데이터 또는 계산 결과를 포함한 청크에 서명하는 행위.

악의적인 검증자는 특정 트랜잭션을 되돌리는 체인 재조직(chain reorganization)을 수행하여, 결과적으로 **이중 지불(double spend)** 을 시도하기 위해 이중 서명을 할 수 있다.

한편, 지분 증명(Proof of Stake) 시스템에서는 악의적 의도 없이도 소프트웨어 설정 오류나 문제로 인해 비의도적 이중 서명이 발생할 수도 있다.

이러한 **우발적 슬래싱(accidental slashing)** 의 위험을 완화하기 위해 NEAR는 “**점진적 슬래싱(Progressive Slashing)**” 방식을 사용한다. 이는 슬래싱되는 스테이크(stake)의 비율이, 해당 에포크(epoch) 내에서 이중 서명 행위를 보인 스테이크의 비율에 3을 곱한 값으로 결정되는 방식이다. 즉, 각 악의적 참여자는 $3 * \text{malicious_stake} / \text{total_stake}$ 만큼 슬래싱된다.

예를 들어, 한 검증자가 특정 에포크에서 전체 스테이킹 토큰의 1%를 보유하고 있다고 가정하자. 총 스테이킹 토큰이 50,000,000 NEAR라면, 해당 검증자는 500,000 NEAR를 스테이킹한 것이다. 이 검증자가 이중 서명을 했고, 다른 검증자는 이중 서명을 하지 않았다면, 이 검증자는 자신의 스테이크의 3%를 잃게 된다. 즉, 485,000 NEAR가 반환되고, 15,000 NEAR가 소각(burned)된다. 만약 한 에포크 내에서 이중 서명된 총 스테이크 비율이 **33%**에 도달한다면(이는 네트워크에 매우 위험한 상태), 관련된 모든 참여자의 스테이크는 전액 슬래싱된다.

잘못된 청크(Invalid Chunk)의 경우, 검증자의 전체 스테이크가 슬래싱된다. 이는 잘못된 청크가 실제로 코드가 수정된 경우에만 발생할 수 있기 때문에, 해당 노드가 **악의적으로 작동했음이 명백**하기 때문이다.

Section 07

기술(Technology)

NEAR의 커뮤니티 운영형 클라우드는 새로운 합의 알고리즘(consensus algorithm)과 확장 가능한 샤딩(sharding) 아키텍처를 통해 높은 수준의 설계 목표를 달성한다.

NEAR 기술의 핵심 요소는 다음과 같다.

- **샤딩(Sharding)**: 시스템은 여러 병렬 샤드에 연산을 분산시켜 수평적이며 사실상 무한한 확장을 달성하도록 설계되었다.
- **합의(Consensus)**: 합의는 모든 샤드에 걸쳐 네트워크를 구성하는 모든 노드 간에 새로운 **Nightshade 알고리즘**을 통해 이루어진다.
- **스테이킹 선택 및 게임 이론(Staking Selection and Game Theory)**: 검증 과정에 참여하기

위해 스테이커(staker)는 보안성이 검증된 무작위(randomized) 절차를 통해 선택된다. 이 과정은 좌석(seats)을 최적화된 방식으로 분배하고, 참여자들이 선의로 행동하도록 인센티브를 제공한다.

- **무작위성(Randomness):** NEAR의 무작위성(Randomness) 생성 방식은 **편향되지 않고, 예측 불가능하며**, 네트워크의 생존성(liveness)이 영향을 받기 전까지 최대 **1/3의 악의적 참여자**를, 그리고 실제 출력값이 조작되기 전까지는 **2/3의 악의적 참여자**까지 견딜 수 있다.

기술 설계 원칙(Technology Design Principles)

NEAR의 전반적 시스템 설계 원칙은 다음과 같은 해석을 통해 기술 설계에 반영된다.

1. **사용성(Usability):** 최종 사용자는 각 상호작용 유형에 대해 가능한 한 낮은 수준의 보안 부담만을 져야 한다. 개발자는 익숙한 언어로 계약을 쉽게 구축, 테스트, 배포할 수 있어야 하며, 최종 사용자에게 현재의 웹 환경과 유사한 경험을 제공할 수 있어야 한다.
2. **확장성(Scalability):** 플랫폼은 사용량이 늘어날수록 무한히 확장할 수 있어야 한다.
3. **단순성(Simplicity):** 시스템의 각 구성 요소는 그 주된 목적을 달성하기 위해 가능한 한 단순하게 설계되어야 한다.
4. **지속 가능한 탈중앙화(Sustainable Decentralization):** 검증 노드로 참여하기 위한 장벽은 가능한 한 낮게 설정되어야 하며, 이를 통해 폭넓은 참여가 가능해야 한다. 또한 먼 미래에 발생하는 개별 트랜잭션은 오늘날의 트랜잭션만큼 안전해야 하며, 이는 그들이 수정하는 가치(value)를 보호하기 위함이다.

요약(Summary)

NEAR는 오늘날 블록체인이 직면한 두 가지 핵심 문제 — **사용성(Usability)** 과 **확장성(Scalability)** — 에 대한 해결책을 제공하는 데 초점을 맞추고 있다.

최종 사용자에게 대한 사용성은 지갑 상호작용을 위한 **점진적 보안 모델(progressive security model)**을 제공하고, 개발자에게 오늘날의 웹 환경과 유사한 사용자 경험을 설계할 수 있는 기회를 제공함으로써 달성된다. 이는 NEAR의 **계약 기반 계정 모델(contract-based account model)**에 의해 프로토콜 수준에서 구현된 유연하고 프로그래머블한 키 관리(key management) 덕분에 가능하다. 이를 통해 **메타 트랜잭션(meta transactions)**, **원자적 계정 전송(atomic account transfers)**, **특정 용도에 잠금된 자금 계정**, 그리고 기타 계정 프로그래머빌리티(account programmability) 및 제한 조건(use restrictions)** 과 같은 기능이 손쉽게 구현된다.

개발자에게 대한 사용성은 브라우저 기반 디버깅(browser-based debugging), 익숙한 프로그래밍 언어(예: AssemblyScript 및 Rust) 지원, 그리고 **계약 사용 리베이트(Contract Reward)** 등의 기능을 프로토콜 차원에서 제공함으로써 달성된다.

확장성은 체인을 잠재적으로 무한한 수의 **병렬 서브체인(subchain)**으로 **샤딩(sharding)**하여 제공된다. 각 샤드는 병렬로 작동함으로써 네트워크 전체의 처리량을 크게 향상시킨다.

성능 특성과 트레이드오프(Performance Characteristics and Tradeoffs)

블록체인 설계에서 자주 언급되는 **트릴레마(trilemma)** 중 하나는, 시스템이 **확장성, 탈중앙화, 보안성**을 동시에 달성할 수 없다는 것이다. NEAR의 샤딩 및 검증자 선택 방식은 이러한 트레이드오프를 완화하여, 확장성과 탈중앙화를 동시에 크게 향상시키면서도 보안 손실을 최소화한다. 또 다른 고전적 트릴레마는 **CAP 정리(CAP theorem)**로, 시스템은 **일관성(Consistency), 가용성(Availability, 즉 “liveness”), 파티션 내성(Partition Tolerance)** 중 오직 두 가지만 달성할 수 있다고 말한다. 이 경우, 파티션 내성은 포기할 수 없으므로 실제 트레이드오프는 일관성과 가용성 간의 문제로 귀결된다.

블록체인 시스템에서 이를 예로 들면, 네트워크가 일주일 동안 두 부분으로 분리되었을 때 발생하는 상황이다. **일관성(consistency)**을 중시하는 시스템은 두 부분이 불일치하지 않도록 네트워크가 복구될 때까지 한쪽(또는 양쪽)을 완전히 중단시킨다. 반면 **가용성(availability)**을 중시하는 시스템(예: 비트코인)은 네트워크의 두 부분이 독립적으로 계속 작동하도록 허용하며, 나중에 네트워크가 재통합되면 한쪽의 작업 내용이 폐기되고 다른 쪽의 기록이 유지된다. NEAR는 시스템 수준에서는 **가용성(availability)**을 우선시하지만, 개별 사용자는 **서명 비율이 50% 이상인 블록만 수락하도록 설정함으로써**, 로컬 수준에서 **일관성(consistency)**을 요구할 수도 있다. 시스템의 성능은 처리되는 트랜잭션 유형과 실제 하드웨어 성능에 따라 달라진다. 단순한 금융 트랜잭션의 경우, 샤드당 초당 **400~2000 TPS(transactions per second)**의 처리량을 달성할 수 있다.

샤딩(Sharding)

현재 확장성(scalability)에 접근하는 방식은 일반적으로 두 가지로 구분된다.

1. 수직 확장(Vertical Scaling):

기존 시스템의 하드웨어 성능을 개선함으로써 달성된다. 블록체인 시스템의 경우, 더 나은 하드웨어를 요구하는 소수의 노드로 네트워크를 구성하는 방식이다. 이는 초기 처리량을 향상시키지만, 향후 확장은 하드웨어 성능 향상 속도(일명 “무어의 법칙”)에 의해 제한된다. 이 방식은 네트워크가 실제 채택 속도(adoption rate)에 맞춰 확장되지 못하게 만든다.

2. 수평 확장(Horizontal Scaling):

시스템에 더 많은 하드웨어를 추가함으로써 달성된다. 블록체인의 경우, 네트워크에 참여하는 노드 수가 증가할수록 그만큼 네트워크 성능도 향상되도록 설계한다. 예를 들어, 여러 샤드에 걸쳐 연산을 병렬화(parallelize)함으로써 처리량을 늘린다.

NEAR는 **수평적 확장(horizontal scalability)**을 제공하기 위해 **샤딩(Sharding)** 방식을 채택한다. 이를 통해 수요 증가에 맞추어 네트워크의 처리 용량을 함께 확장할 수 있다.

크로스체인 및 크로스샤드 통신(Cross-Chain and Cross-Shard Communication)

단일 체인의 여러 샤드 간이든, 여러 체인 간이든 **크로스체인 통신(cross-chain communication)** 에서 가장 어려운 과제 중 하나는 다른 체인에서 들어오는 트랜잭션이 유효한지를 판별하는 것이다. 크로스체인 트랜잭션을 검증하는 방식에는 세 가지 주요 접근법이 있다.

1. 이중 검증(Dual Validation):

수신 체인의 검증자들이 송신 체인에서도 동시에 검증을 수행하는 방식이다. [Quarkchain](#)이 이 방식을 사용한다. 단점은 검증자가 이 방식에서는 확장성이 떨어진다는 점이다.

2. 트랜잭션 신뢰(Trust the Transaction):

수신된 트랜잭션이 유효하다고 가정하는 방식이다. 예를 들어 [Cosmos](#)에서는 메인 허브에 복사된 트랜잭션은 되돌릴 수 없는 것으로 간주한다. 각 경제권의 총 토큰 수를 추적함으로써 새로운 토큰이 생성되는 것은 방지할 수 있지만, 이론적으로는 당사자 간 잘못된 송금(예: 다른 사용자의 토큰 탈취)이 발생할 수 있다.

3. 비콘 체인 + 롤백(Beacon Chain w/ Rollback):

비콘 체인(beacon chain)은 소수의 검증자 하위 집합을 통해 다른 모든 체인의 상태 전이를 검증하며, 문제가 발견될 경우 전체 체인을 롤백한다. 원자성(atomicity)을 달성하려면 이러한 롤백이 반드시 일어나야 하지만, 매우 드물게 발생해야 하며 즉시 감지되어야 한다.

NEAR는 세 번째 접근 방식을 채택한다. 즉, **적응적 공격자(adaptive adversary)** 가 하루 안에 특정 샤드의 검증자들을 부패시킬 수 없다는 가정하에, 각 샤드의 검증자 집합을 매일 회전시켜 보안을 강화한다. 물론, 이론적으로는 하루 안에 특정 샤드의 검증자 집합을 부패시키는 것이 매우 어렵지만 가능할 수도 있다.

이를 보완하기 위해, 다른 프로토콜들은 훨씬 더 빠르게(예: 몇 분마다) 회전하는 소규모 위원회(committee)를 사용하여 여러 샤드를 검증한다. 이 소규모 위원회는 각 샤드의 전체 상태(state)를 다운로드하지 않고 검증을 수행해야 하므로, 트랜잭션으로 실제로 영향을 받은 상태만 전달받는다. 그러나 상태 데이터를 매번 전송하는 것은 비효율적이다 — 단일 트랜잭션이 한 번에 100MB의 상태를 변경할 수도 있기 때문이다.

이 지점에서 **Nightshade 샤딩 접근법**이 등장한다.

나이트쉐이드(Nightshade)

Nightshade는 기존의 샤딩 추상화 방식을 수정하여, 모든 샤드가 하나의 블록을 함께 구성한다고 가정한다. 이 블록은 각 샤드가 해당 블록 높이에서 자신의 “청크(chunk)”를 생성했는지 여부와 상관없이 일정한 주기로 생성된다. 즉, 각 샤드의 청크는 특정 블록 내에 **존재하거나 존재하지 않거나** 둘 중 하나이다.

올바른 포크(fork)를 결정하기 위한 **포크 선택 규칙(fork choice rule)** 이 필요하다. 이 규칙은 아직 개발 중이지만, **LMD Ghost**와 유사한 형태를 띠 것으로 예상된다. 또한 특정 청크와 블록에 대해 수신된 검증자 서명(attestation)의 가중치(weight)도 포함된다.

각 블록은 하나의 검증자에게 생산이 할당된다. 이 검증자는 해당 블록 기간 동안 제공받은 모든 청크를 모아 하나의 블록으로 조립해야 한다. 이 검증자 할당은 기존 검증자 집합(예: 100명) 내에서 순환(rotating) 방식으로 진행된다. 이 블록 리더는 새로운 트랜잭션을 직접 수락하지 않고, **오직 청크만을 수집**한다.

각 샤드 및 주기(period)마다 하나의 검증자가 해당 샤드의 청크를 생성하도록 할당된다. 만약 해당 검증자가 부재할 경우, 그 샤드는 해당 기간 동안 일시 중단(stall)된다. 각 샤드는 메인 검증자 풀에서 추출된 자체 소규모 검증자 풀을 가지고 있으며(예: 4명), 이들 간에도 블록 리더와 동일한 방식으로 리더 역할이 순환된다. 따라서 특정 기간 동안 한 검증자가 부재하여 청크 생성이 일시 중단되더라도, 다음 주기에는 다른 검증자가 참여하여 체인의 작동을 이어갈 가능성이 높다.

NEAR의 샤딩 설계에 대한 더 자세한 내용은 [Nightshade 논문](#)에서 확인할 수 있다.

히든 검증자(Hidden Validators)

추가적인 보안을 제공하기 위해 NEAR는 **히든 검증자(Hidden Validators)**를 사용한다. 이들은 각 샤드별로 평균 약 100명의 검증자로 구성된 소규모 위원회이며, 각 청크(chunk)를 검증한다. 이들의 할당 정보는 블록체인 상에 공개되지 않으며, 각 검증자는 **검증 가능한 난수 함수(Verifiable Random Function, VRF)**를 통해 무작위로 선택된 샤드 ID 세트를 추출함으로써 자신이 어떤 샤드를 검증해야 하는지 **개별적으로** 파악한다.

이 방식 덕분에, 각 검증자는 자신이 검증해야 할 샤드를 알고 있지만, 외부 공격자가 이들을 부패시키려면 모든 샤드에 걸쳐 **전체 검증자의 상당 비율**을 매수해야 하므로 공격이 사실상 불가능해진다.

또한 특정 블록에 지정되는 히든 검증자의 수도 무작위로 결정된다. 이는 공격자가 처음부터 몇 명의 히든 검증자를 부패시켜야 공격이 성공하는지조차 예측할 수 없게 만들어 공격 가능성을 더욱 낮춘다. 따라서 공격자가 의도적으로 특정 샤드를 노리고 “피셔맨(Fisherman)”이 자신을 찾아오기를 기다리는 형태의 공격도 불가능하게 된다.

검증의 특성상, **단일 히든 검증자라도 청크가 잘못되었음을 증명(fraud proof)**할 수 있다. 이러한 “사기 증명(fraud proof)”은 네트워크의 자동 슬래싱(slash) 및 롤백(rollback) 절차를 촉발할 수 있다.

히든 검증자를 포함한 소규모 샤드별 위원회는 매 에포크($1/2$ 일)마다 블록 및 청크 생성자와 동일한 풀에서 무작위로 선출된다. 예를 들어, 샤드당 100개의 좌석(seats)이 있고 100개의 샤드가 존재한다면 전체 좌석 수는 10,000개이다. 이 중 100개는 청크 생성자(chunk producers)에게 할당되고, 나머지는 히든 검증자로 배정된다.

피셔맨(Fishermen)

각 샤드의 보안을 담당하는 히든 검증자 외에도, 모든 노드 운영자는 **피셔맨(Fisherman)**으로서 무허가(permissionless) 방식으로 참여할 수 있다. 피셔맨 노드는 히든 검증자와 동일하게 사기 증명(fraud proof)을 제출할 수 있으며, 이를 통해 슬래싱 및 롤백 절차를 시작할 수 있다.

이 메커니즘은, 설령 공격자가 전체 히든 검증자 풀을 부패시키는 데 성공하더라도, 독립적인 피셔맨 노드가 이를 탐지할 가능성을 남기므로 공격 시도를 **강력하게 비인센티브화(disincentivize)**한다.

게으른 검증자 방지(Preventing Lazy Validators)

검증자 관련 잠재적 문제 중 하나는, 검증자가 “게으르게(lazy)” 행동하는 것이다. 각 블록이 생성될 때마다, 검증자는 새 청크를 수신하고, 새로운 상태(state)를 다운로드하며, 해당 블록을 검증해야 한다. 그러나 일부 검증자는 다른 검증자가 사기 증명(fraud proof)을 제출하기 전까지 아무 작업도 하지 않고 기다리다가, 문제가 감지되면 그제야 검증을 수행하고 자신도 증명을 제출하려고 시도할 수 있다. 이 경우 체인은 검증자에게 보상을 지급하지만 실제로 의미 있는 작업은 수행되지 않는 상황이 된다.

이를 방지하기 위해 NEAR는 커밋-리빌(commit-reveal) 메커니즘을 도입한다. 즉, 검증자는 먼저 자신이 청크가 유효(valid)하다고 생각하는지, 혹은 유효하지 않다고 판단하는지에 대한 결정을 커밋(commit)해야 하며, 이후에 이를 공개(reveal)해야 한다. 이로 인해 검증자는 실제 검증 작업을 수행할 인센티브를 갖게 되며, 잘못된 청크를 검증하지 않거나 누락하면 자신의 스테이크가 슬래싱되므로, 부실한 행동이 억제된다.

데이터 은폐 방지(Preventing Data Hoarding)

또 다른 문제는 청크 생성자가 악의적으로 행동하여 청크를 손상(corrupt) 시키고, 히든 검증자에게 필요한 데이터를 제공하지 않음으로써 검증 또는 사기 증명(fraud proof)을 불가능하게 만드는 경우이다.

이를 해결하기 위해 NEAR는 청크 생성자가 자신이 생성한 청크를 다른 샤드의 청크 생성자들에게 삭제 부호화(eraser coded) 형태로 전송하도록 요구한다. 이 코드 덕분에, 전체 참여자 중 단 16%만으로도 해당 청크를 복원할 수 있으며, 히든 검증자들이 이를 검증할 수 있다. 만약 (악의적인) 청크 생성자가 이러한 삭제 부호화를 제공하지 않았다면, 다른 청크 생성자는 해당 청크의 일부를 보유하지 않으므로 그 위에 새로운 블록을 빌드하거나 서명(attest)하지 않는다. 결과적으로 포크 선택 규칙(fork choice rule)은 최소 50% 이상의 참여자가 일부 데이터를 보유한 체인을 선택하게 된다.

악의적 행동을 억제하기 위해, 삭제 부호화된 청크의 일부는 의도적으로 “지뢰(land mine)”로 설정된다 (즉, 유효하지 않은 조각을 포함). 만약 어떤 검증자가 해당 지뢰가 포함된 블록에 서명한 것으로 확인된다면, 이는 쉽게 증명 가능하며, 해당 검증자는 즉시 슬래싱된다. 따라서 각 기간마다 악의적 행위자가 슬래싱될 확률이 존재하며, 이는 악의적 행동에 대한 매우 강력한 억제력(deterrent)으로 작용한다.

무작위성(Randomness)

블록체인에서의 무작위성(Randomness)은 다음과 같은 특성을 가져야 한다.

1. 편향 불가(Unbiasable)
2. 예측 불가(Unpredictable)
3. 생존성(Liveness) — 일부 노드가 오프라인 상태이거나 악의적 참여자가 있어도 작동해야 함

이러한 무작위성을 생성하기 위한 여러 접근법이 존재한다.

1. RANDAO — 예측 불가능하지만 편향될 수 있음. 생존성은 기본 합의 프로토콜에 의존한다.

2. **RANDAO + VDF (Verifiable Delay Function)** — 예측 불가능하고 편향 불가능하며 생존성을 갖춤. 그러나 실제 구현 시 ASIC(전용 하드웨어) 공격에 동시에 저항하기가 어렵다.
3. **임계값 서명(Threshold Signatures)** — 예측 불가능하고 편향 불가능하며 생존성을 갖춤. 그러나 특정 방식으로 개인 키를 생성해야 하므로 메커니즘이 복잡하다. 현재 활발한 연구 분야이다.
4. **RandShare** — 예측 불가능하고 편향 불가능하며 생존성을 갖춤. 그러나 네트워크 메시지 수가 $O(n^3)$ 에 달해, 참여자 수 n 이 많을 경우 매우 비효율적이다. 또한 악의적 참여자가 $1/3$ 을 초과할 경우 편향이 가능해진다.

NEAR의 접근 방식은 **예측 불가능하고, 편향 불가능하며, 생존성을 갖춘다**. RandShare와 달리, NEAR는 악의적 참여자가 전체의 $2/3$ 에 달하기 전까지 편향되지 않는다. 임계값 서명 방식보다 훨씬 단순하며, RANDAO+VDF와 달리 ASIC 공격에도 취약하지 않다. 자세한 작동 방식은 [NEAR Randomness 논문](#)에서 확인할 수 있다.

거버넌스(Governance)

거버넌스는 프로토콜이 **어떻게 업데이트되는지(기술적 거버넌스, Technical Governance)**와 **리소스가 어떻게 배분되는지(자원 거버넌스, Resource Governance)**를 정의한다. 기술적 거버넌스는 일반적으로 버그 수정, 시스템 매개변수 업데이트, 그리고 프로토콜의 근본적인 기술에 대한 대규모 변경 사항을 포함한다. 자원 거버넌스는 일반적으로 커뮤니티 주도의 자금(예: 재단을 통해 배정된 보조금) 배분을 포함한다.

기술적 거버넌스는 전 세계의 수천 명에 이르는 독립 노드 운영자 간 조정이 필요하므로 특히 복잡하다. 각 노드는 네트워크의 최신 버전에 참여하기 위해 업그레이드 과정을 거쳐야 하며, 그렇지 않으면 별도의 체인을 실행하려는 시도를 하게 될 수 있다. 따라서 업그레이드 과정은 원활해야 하며, 영향을 받는 노드들이 결정 과정에 신뢰를 가질 수 있어야 한다.

많은 프로토콜들은 거버넌스의 **의사결정 과정**을 “오프체인(off-chain)”에서 수행한다. 즉, 텍스트 채널, 직접 회의, 전화 통화를 통해 주요 이해관계자나 그 대표들이 최적의 조치를 논의하고 결정한다. 이 방식은 인간의 유연한 의사소통을 통해 주요 문제를 해결할 수 있지만, 강력한 인물이나 비공식적 권력 구조(soft power)에 따른 영향력을 받기 쉽다는 단점이 있다.

다른 프로토콜들은 “온체인(on-chain)” 거버넌스에 의존한다. 즉, 프로토콜의 주요 자원을 보유한 참여자(예: 토큰 보유자)가 **온라인 투표 메커니즘**을 통해 명시적으로 결정을 내린다. 이 방식은 의사결정과 배포 과정을 명확히 정의할 수 있는 장점이 있지만, 모든 사안을 매우 구체적으로 명시해야 한다는 한계가 있다. 또한 일부 의사결정에서는 인간의 “상식(human common sense)”이 결여되어 특정 공격에 취약할 수 있으며, 이는 오프체인 방식에서는 발생하지 않을 문제일 수도 있다.

거버넌스 설계 원칙(Governance Design Principles)

다음은 NEAR의 핵심 설계 원칙이 거버넌스에 적용되는 방식이다.

1. 사용성(Usability):

거버넌스 프로세스는 명확하고 이해하기 쉬워야 한다. 참여 및 투표(가능한 경우)를 위한 메커니즘은 단순하고 직관적이어야 한다. 거버넌스는 빠르고 효율적으로 결정을 내리고 실행할 수 있어야 한다. 이해관계자 커뮤니티는 의사결정의 정당성을 지지할 만큼 충분한 발언권을 가져야 하며, 플랫폼을 이탈하거나 포크하지 않도록 해야 한다.

2. 확장성(Scalability):

거버넌스는 플랫폼의 범위와 복잡성이 커지고, 이해관계자의 다양성이 증가하며, 참여 폭이 확장됨에 따라 함께 확장 가능해야 한다.

3. 단순성(Simplicity):

가장 견고한 프로세스는 대체로 가장 단순한 프로세스이다. 따라서 좋은 거버넌스는 과도한 복잡화(overengineering)를 피하고, 종종 인간 간의 직접적인 의사소통이 가장 효율적이라는 점을 인정해야 한다.

4. 지속 가능한 탈중앙화(Sustainable Decentralization):

거버넌스는 플랫폼 내의 모든 이해관계자에게 참여 기회를 제공해야 하며, 시간이 지나도 특정 집단에 의해 장악(capture)되지 않도록 회복력을 가져야 한다.

거버넌스 설계에서 중요한 점은 **효율성(efficiency)**과 **탄력성(resiliency)**의 균형이다. 기술 플랫폼이 지속적으로 진화하여 이해관계자에게 최고의 가치를 제공하려면 효율적인 의사결정과 실행이 필요하지만, 동시에 특정 집단이 장기적으로 플랫폼을 장악하지 못하도록 보장해야 한다.

요약(Summary)

NEAR의 거버넌스는 프로토콜의 효율적인 개선을 가능하게 하는 동시에, 커뮤니티가 충분한 발언권과 감시 기능을 통해 프로토콜의 독립성을 유지하도록 설계되었다. 장기적으로는 커뮤니티 주도의 혁신을 효과적인 의사결정 및 실행과 결합하고, 네트워크의 주요 이해관계자 그룹들이 모두 적절히 대표될 수 있도록 하는 것이 목표이다.

예를 들어, NEAR 커뮤니티는 초기 단계에서 토큰 보유자, 검증자, 애플리케이션 개발자, 프로토콜 개발자, 커뮤니티 리더 등을 포함한다. 이들 각 이해관계자는 주요 사안에 대해 서로 다른 관점과 의견, 그리고 입력값(input)을 가진다.

적절한 대표성(representation)을 보장한다는 것은, 결정이 충분한 토론과 숙의를 거쳐야 함을 의미하며, 이는 통제되지 않을 경우 프로토콜의 진화 속도를 늦출 수도 있다. 따라서 효율적인 실행을 유지하기 위해, **핵심 프로토콜 코드의 참조 구현(Reference Implementation)**을 유지할 고도로 전문화된 주체가 필요하다. 이 주체는 **참조 유지자(Reference Maintainer)**라고 불리며, 커뮤니티의 감독하에 선출되어야 한다.

초기에는 NEAR 재단(NEAR Foundation)이 독립적 비영리 기관으로서 거버넌스 활동을 조정한다. 재단의 사명은 생태계의 장기적 유용성을 개선하는 것이다. 이 활동에는 참조 유지자에 대한 감독, 거버넌스 조정 도구 구축 지원, 일부 토큰 배분 관리, 커뮤니티 기반 거버넌스의 기반 조성 등이 포함된다.

기술 거버넌스(Technical Governance)

탈중앙화 네트워크로서, 단일 주체가 NEAR 네트워크 전체에 변화를 강제할 수는 없다. 참조 코드베이스(reference code base)에 대한 핵심 기여자들의 변경 사항은 네트워크를 운영하는 각 노드에 의해 개별적으로 수용되어야 한다.

그러나 참조 코드베이스에 변경 사항을 적용하는 핵심 과정은 이해해야 한다. 이 과정은 커뮤니티의 의지를 가장 잘 반영하며, 결과적으로 네트워크 노드들이 해당 변경을 수용할 가능성이 높기 때문이다.

NEAR의 거버넌스는 **참조 유지자(Reference Maintainer)**를 정의한다. 이는 NEAR 네트워크의 기술적 업그레이드를 담당하는 주체이다. 이 유지자는 **참조 구현(Reference Implementation)**을 관리하고, 명세(specification)에 대한 개선을 지속적으로 제안하는 역할을 맡는다. 모든 주요 릴리스는 커뮤니티의 토론 및 **비토(veto) 절차**(2주간의 챌린지 기간)를 거쳐 보호되며, 작은 버그 수정은 빠르게 배포되어 노드 운영자에게 제공될 수 있다.

초기에는 참조 유지자가 **재단 이사회(Foundation Board)**에 의해 선출되며, 이사회가 교체를 의결할 때까지 임기를 유지한다. 시간이 지나면 유지자에 대한 감독은 커뮤니티를 대표하는 **선거 과정**을 통해 수행될 예정이다.

자원 거버넌스(Resource Governance)

네트워크 자체가 프로토콜 재무금고(Protocol Treasury)에 제공하는 자원은 **NEAR 재단(NEAR Foundation)**에 의해 관리되고 분배된다. 이 재단은 독립적으로 운영되며, 프로토콜 생태계의 지속적인 건강성과 발전에 가장 도움이 된다고 판단되는 프로젝트와 활동에 대해 **구조적이고 투명한 자금 지원**을 제공한다. 여기에는 **참조 유지자(Reference Maintainer)**와 같은 기술적 프로젝트뿐 아니라, 커먼스(the commons)와 커뮤니티 전반을 지원하는 비기술적 프로젝트나 이니셔티브도 포함된다.

프리티미브(Primitives)

새로운 컴퓨팅 패러다임이 등장할 때마다, 그것이 어떻게 가장 효과적으로 활용될 수 있는지, 그리고 미래의 혁신에 어떤 의미를 가지는지에 대한 불확실성이 뒤따른다. 이번에도 예외는 아니다.

NEAR와 같은 블록체인 기반 애플리케이션 플랫폼은 **컴퓨트(compute)**와 **스토리지(storage)**라는 두 가지 기존 클라우드 서비스를 **신뢰 불필요(trustless)**하고 **허가 불필요(permissionless)**한 방식으로 결합한다. 이러한 조합은 완전히 새로운 형태의 **프리티미브(primitive)**를 만들어내며, 이를 통해 새로운 애플리케이션, 가치 사슬(value chains), 비즈니스를 구축할 수 있다.

이처럼 새로운 기술이 등장했을 때는, 우선 근본적인 수준에서 이 기술이 **무엇을 할 수 있는지**, 그리고 **무엇을 해서는 안 되는지**를 이해하는 것이 중요하다. 새로운 프리미티브와 가치 사슬이 만들어내는 가능성의 우주는 기술 도입 초기 단계에서 완전히 예측할 수 없다. 예를 들어, 2000년대 초반 첫 스마트폰이 등장했을 때, 주머니 속의 카메라와 GPS가 세상을 어떻게 바꿀지 예측한 사람은 거의 없었다.

이후 섹션에서는 NEAR 기술이 **어떤 분야에 적합한지**, **어떤 분야에는 적합하지 않은지**, 그리고 현재 이 기술이 가능하게 하는 다양한 프리미티브를 탐구한다. 또한 블록체인을 언제 적용해야 하고, 언제 피해야 하는지를 이해할 수 있는 명확한 **사고 모델(mental model)**을 제공한다. 미래 전망에 대한 논의는 본 백서의 결론 부분에서 다룬다.

<https://near.org/papers/the-official-near-white-paper/>

기술이 적합하지 않은 영역(What the Technology is Not Best At)

프리미티브에 대해 논의하기 전에, 먼저 이 기술이 **가장 잘 맞지 않는 분야**를 인식함으로써 흔히 존재하는 오해를 바로잡는 것이 중요하다.

NEAR와 같은 **커뮤니티 운영형 클라우드(community-operated cloud)**는 기존의 단일 컴퓨트 및 스토리지 솔루션에 비해 저렴하지도, 빠르지도 않다. 이는 구조적으로 불가피한 일이다 — 이러한 네트워크의 핵심 이점은 연산과 저장의 **중복성(redundancy)**을 활용해 강력한 보안을 제공한다는 데 있기 때문이다.

이 원리를 가장 단순한 수준에서 이해하려면, 이러한 네트워크가 여러 개의 독립 노드로 연산 및 저장을 **병렬 분산(agggregation)**하는 방식을 살펴보면 된다. 예를 들어, 네트워크가 하나의 샤드(shard)로 구성되어 있고, 그 안에 100개의 노드가 각각 독립적인 하드웨어에서 병렬로 작동한다면, 네트워크 상에서의 연산 비용은 단일 하드웨어에서 연산하는 비용의 **최소 100배 이상**이 되며, 노드 간 통신 시간이 추가되므로 **속도는 오히려 느려질** 수밖에 없다.

마찬가지로, 네트워크의 저장 용량은 이론적으로는 무한하지만, 실제로는 **새로운 검증 노드(validating nodes)를 추가할 수 있는 속도와 비용**에 의해 제한된다. 각 샤드의 저장 용량은 새로운 검증자가 네트워크에 참여하여 새로운 샤드로 동기화할 수 있을 만큼으로 고정되어 있다. 샤드 내의 모든 검증자가 저장소를 복제하기 때문에, 네트워크의 저장 용량을 늘리려면 **새로운 샤드가 생성되어야 하며**, 그에 따라 **새로운 검증자 집합이 추가되어야 한다**.

앞서 예시를 기준으로 보면, 네트워크에 새로운 저장 용량을 추가하기 위해서는 또 다른 100명의 검증자를 참여시켜야 하며, 이는 완전히 새로운 검증자일 수도 있고, 기존 검증자가 더 많은 샤드를 담당하도록 확장하는 것일 수도 있다. 따라서 네트워크의 저장 비용은 이러한 현실을 반영해야 하며, 항상 단일 저장 장비를 운영하는 것보다 **몇 배 이상 높은 비용**이 들 수밖에 없다. 물론 저장 기술의 발전은 필요한 저장 공간을 최적화할 수 있지만, 이러한 **경제적 구조의 본질적 제약**을 제거할 수는 없다. 결과적으로, 블록체인 기반 시스템은 **아마존 AWS나 구글 클라우드(Google Cloud)**와 같은 중앙화된 클라우드 컴퓨팅 시스템보다 전통적인 연산이나 저장 용도에서 더 빠르거나 저렴할 수 없다는 점이 분명하다. 즉, **성능 및 비용 최적화가 핵심인 애플리케이션은 NEAR 플랫폼의 초기 구현 단계에서 가장 적합한 사용 사례가 아니다**.

기술의 새로운 이점(New Benefits of the Technology)

플랫폼이 가능하게 하는 구체적인 프리미티브를 살펴보기 전에, 커뮤니티 클라우드를 통해 **허가 불필요(permissionless)** 하고 **신뢰 불필요(trustless)** 한 방식으로 컴퓨트와 스토리지를 결합했을 때 제공되는 근본적인 이점을 이해하는 것이 중요하다. 이러한 개념적 이해는 단순히 “블록체인이 기존 클라우드 아키텍처보다 더 빠르거나 저렴하게 기존 작업을 수행할 수 있는가?”라는 질문을 넘어서, “이 기술이 어떤 **새로운 사용 사례(new use cases)** 를 가능하게 하는가?”라는 관점으로 나아가게 한다.

고가치 자산을 위한 탈중앙화(Decentralization for High Value Assets)

탈중앙화(decentralization)는 시스템이 손상되기 위해 **얼마나 많은 참여자가 부패해야 하는가**를 나타내는 축 위의 **스펙트럼**으로 이해할 수 있다. 이 스펙트럼의 한쪽 끝에는 오늘날 대부분의 시스템(예: 클라우드 컴퓨팅 및 스토리지 솔루션)이 있으며, 이들은 단 하나의 주체가 접근 키(access key)를 보유함으로써 해당 시스템의 데이터를 임의로 수정할 수 있으므로 **중앙화 계수(centralization factor)**가 1이다. 반대쪽 끝에는 완전히 탈중앙화된 시스템이 있으며, 이 경우 기본 데이터를 손상시키려면 수십, 수백, 혹은 수천 명의 참여자를 부패시켜야 한다.

오늘날의 많은 블록체인 기반 시스템은 실제로 상당히 중앙화되어 있다. 이는 설계상의 이유이거나, 스테이킹 풀이나 위임(delegation) 메커니즘이 **소수의 강력한 참여자**를 형성하게 되는 인센티브 구조 때문이거나, 혹은 거버넌스 절차가 쉽게 장악(capture)될 수 있기 때문이다. 앞서 언급했듯이, 탈중앙화가 제공하는 **중복성(redundancy)** 은 추가 비용을 발생시키며, 따라서 탈중앙화가 제공하는 이점이 그 비용을 상쇄할 만큼 충분히 중요한 사용 사례에서 가장 큰 효용을 발휘한다. 오늘날 애플리케이션에 저장되는 데이터의 대부분은 **저가치(high volume, low value)** 데이터이다. 그러나 탈중앙화는 **검열, 도난, 변조, 조작 등에 취약한 고가치(high value)** 데이터의 경우에 가장 중요하다. 따라서 **디지털 화폐, 신원(identity), 자산 소유권(asset ownership)** 을 나타내는 데이터가 NEAR 플랫폼에서 저장될 때 가장 큰 이점을 얻는다.

공유 글로벌 데이터 레이어(A Shared Global Data Layer)

NEAR 플랫폼은 애플리케이션들이 **공유 데이터 풀(shared data pool)** 에 접근할 수 있도록 하여, 여러 애플리케이션이 동일한 상태(state)를 쉽게 공유할 수 있게 한다. 이는 각 애플리케이션이 일반적으로 자체 독점 데이터베이스에 데이터를 저장하고, 서로 간의 데이터 통신이 어려운 **기존 웹 애플리케이션 구조**와는 다르다. 애플리케이션 간에 공유되는 상태(state)는 디지털 화폐, 신원, 자산 등 앞서 언급된 어떤 데이터 유형이든 될 수 있다. 이 데이터는 기본적으로 **암호학적으로 보호(cryptographically secured)** 되며, 사용자의 명시적 권한을 가진 애플리케이션만 해당 데이터를 수정할 수 있다. 사용자는 자신의 데이터를 **실질적으로 소유**하므로, 제3자 애플리케이션의 허가 없이도 이를 수정하거나 이전(transfer)할 수 있다. 이는 NEAR가 단순히 **금융 자산과 같은 고가치 데이터**를 저장할 수 있을 뿐 아니라, 사용자와 애플리케이션 간에 이러한 자산을 **자유롭게 전송**할 수 있게 함을 의미한다. 이러한 기능은 오늘날의 중앙화된 플랫폼에서는 불가능하다.

또한, 데이터는 여전히 암호화되어 보안 및 프라이버시를 보장할 수 있다는 점도 주목해야 한다.

개발자가 “무료로” 얻는 것(What Developers Get “For Free”)

핵심 이점 외에도 NEAR는 개발자에게 일반적으로 수개월의 설정이 필요한 분산 아키텍처에 대한 즉각적인 접근성을 제공한다. 여기에는 파티션 내성 네트워킹(partition-resilient networking),고가용성 데이터베이스(high-availability database), 그리고 전 세계에 분산된 엔드포인트가 포함된다. 또한 개발자는 현재의 웹 환경에서는 구현에 상당한 개발 노력이 필요한 다양한 프리미티브(primitives)에 쉽게 접근할 수 있다. 예를 들어, 개발자는 민감한 사용 사례에 필요한 암호학적 프리미티브(cryptographic primitives)에 네이티브로 접근할 수 있으며, 모든 플랫폼 사용자가 공통적으로 이용할 수 있는 싱글사인온(SSO, Single Sign On)을 애플리케이션에서 손쉽게 활용할 수 있다. 이를 통해 사용자는 새로운 애플리케이션을 시도할 때 거의 마찰 없는(onboarding frictionless) 경험을 할 수 있다.

네이티브 프리미티브(Native Primitives)

이제 컴퓨트와 스토리지를 결합해 하나의 탈중앙화되고 공유된 데이터 레이어를 형성했을 때의 고차원적 이점을 살펴봤으니, 이를 통해 어떤 새로운 프리미티브가 가능해지는지 살펴보자. 프리미티브는 사용 사례(use cases)의 기본적인 구성 요소(building blocks)이다. 이러한 프리미티브는 다음과 같은 범주로 구분된다.

1. **자산(Assets):** 화폐에서 데이터에 이르기까지 모든 형태의 자산이 이제 디지털 네이티브(digitally native) 형태로 존재하며, 이는 검증 가능한 고유성과 개인 소유권을 가지며 완전히 프로그래밍 가능하다.
2. **계정(Accounts):** 생태계의 모든 참여자는 자신만의 계정을 가지며, 이를 통해 자산을 안전하게 보관하고, 애플리케이션에 대한 신원을 쉽게 검증하며, 시간이 지남에 따라 평판(reputation)을 축적할 수 있다.
3. **트랜잭션(Transactions):** 자산이 디지털 네이티브이며 계정이 글로벌 풀(global pool)의 일부이기 때문에, 당사자 간의 프로그래머블 트랜잭션은 단순하고, 저렴하며, 안전하고, 거의 즉시 처리된다.
4. **검증(Verification):** NEAR의 저장소는 변경 불가능한(publicly immutable) 공개 원장(public ledger)이므로, 플랫폼에 저장된 데이터와 코드는 시점(timing)과 내용(content) 모두에 대해 공개적으로 검증 가능하다.

이제 다음 섹션에서 이러한 각 프리미티브를 더 깊이 탐구한다.

자산 프리미티브(Asset Primitives)

자산은 이제 디지털 네이티브하며, 검증 가능한 고유성을 갖추고 완전히 프로그래밍 가능하다. 이는 기존 자산에 디지털화의 이점을 제공하거나, 완전히 새로운 범주의 자산을 창출하는 데 사용될 수 있다. 예를 들어, 화폐가 과거에는 단일 차원의 상품이었지만, 이제는 완전히 프로그래밍 가능하다 — 발행 조건부터 사용 조건에 이르기까지 모든 요소가 자산 자체에 직접 통합될 수 있다.

자산 프리미티브에는 다음이 포함된다.

1. **프로그래밍 가능한 소유권(Programmatic Ownership):** 각 계정은 자신이 보유한 화폐, 디지털

상품, 데이터(예: 신원 정보와 같은 실물 자산의 표현)에 대한 검증 가능한 통제권을 가지며, 이를 프로그래밍적으로 결정하거나 분할할 수 있다.

2. **디지털 고유성(Digital Uniqueness)**: 디지털 자산은 100% 고유할 수 있으며, 특정 대체 가능한 토큰(fungible token)부터 완전히 대체 불가능한 토큰(non-fungible token, NFT)에 이르기까지 어떤 것을 표현할 수도 있다.
3. **프로그래밍 가능한 자산(Programmable Assets)**: 자산의 생성, 진화, 소멸을 프로그래밍 방식으로 제어할 수 있다.

계정 프리미티브(Account Primitives)

생태계의 모든 행위자(actor) — 인간, 계약(contract), 또는 장치(device) — 는 최상위 계정(top-level account)을 가진다. 각 행위자를 플랫폼의 일급 시민(first-class citizen)으로 취급함으로써, 기존 세계의 신원(identity) 개념과 자율적 혹은 반자율적 행위자 간의 새로운 상호작용 방식을 모두 가능하게 한다. 계정 프리미티브에는 다음이 포함된다.

1. **자율성(Autonomous)**: *체인 상의 모든 것은 하나의 계정에 할당된다. 계정은 사람, 기업, 애플리케이션, 또는 사물(예: 냉장고)일 수도 있다. 모든 계정은 구분 없이 일급 시민으로 취급된다.*
2. **싱글사인온(SSO, Single Sign-on)**: 하나의 계정으로 모든 애플리케이션 및 체인과 연동되는 다른 시스템 전반에서 사용할 수 있다.
3. **평판/이력(Reputation/History)**: 각 계정의 트랜잭션 이력은 서비스 전반에서 활용 가능한 평판(reputation)을 형성한다.

트랜잭션 프리미티브(Transaction Primitives)

자산(Asset)과 계정(Account) 프리미티브가 동일한 공유 데이터 풀(shared data pool) 안에서 제공되기 때문에, 이들 요소 간에 디지털 네이티브 환경에서만 가능한 **매끄럽고 자동화된 상호작용(seamless interactions)**을 손쉽게 구현할 수 있다. 가장 흔히 언급되는 사용 사례는 **허가 불필요한(peer-to-peer)** 금전 거래이지만, 이는 모든 형태의 디지털 자산 전송에도 동일하게 적용된다. 트랜잭션 프리미티브에는 다음이 포함된다.

1. **직접 전송(Direct)**: 특정 애플리케이션의 코드나 권한 승인 없이 계정 간 직접 전송이 가능하므로, 개인 간 거래(P2P 마켓플레이스)나 자산 이전이 가능하다.
2. **즉시성(Instant)**: 금융 및 디지털 자산 거래는 몇 초 만에 확정(finality)에 도달하며, 긴 확인 대기 기간이 필요하지 않다.
3. **소액성(Micro)**: 무시할 수준의 낮은 수수료로 인해, 고빈도 거래나 소액 전송이 기존 금융 인프라보다 훨씬 실현 가능해진다.
4. **조건부(Conditional)**: 스마트 계약은 거래에 논리를 추가할 수 있으며, 예를 들어 조건부 에스크로(escrow) 또는 시간 기반 지급(time-based release)과 같은 기능을 손쉽게 구현할 수 있다.

검증 프리미티브(Verification Primitives)

NEAR에서 데이터를 저장하는 데 사용되는 변경 불가능한 공개 원장(immutable public ledger)은 과거에 발생한 사건의 **검증 가능한 기록(verifiable record)** 과 특정 애플리케이션의 백엔드에서 실행되는 코드의 **검증 가능한 저장소(verifiable repository)** 를 모두 생성한다. 이 구조는 작은 규모의 예시(예: 주사위 게임이 실제로 무작위성을 사용하는가?)에서부터 큰 규모의 예시(예: 공급망 감사(audit trail) 생성)에 이르기까지 다양한 방식으로 활용될 수 있다. 검증 프리미티브에는 다음이 포함된다.

1. **체크포인팅(Checkpointing)**: 암호학적 타임스탬프를 사용하여 특정 시점의 상태(state)를 검증 가능한 체크포인트로 저장할 수 있으므로, 애플리케이션은 과거 활동의 진위 여부나 발생 시점을 손쉽게 검증할 수 있다.
2. **프로세스 무결성 검증(Verify Process Integrity)**: 플랫폼에 배포된 애플리케이션 코드는 현재의 서버 기반 코드(server-side code)와 달리 검증 가능한 방식으로 실행 무결성을 확인할 수 있다.

조합 프리미티브(Combinatorial Primitives)

플랫폼 기술이 네이티브로 제공하는 저수준 프리미티브를 살펴보는 것은 어렵지 않지만, 가장 흥미로운 가능성은 여러 프리미티브를 조합해 **고차원 프리미티브(higher-level primitives)** 를 만드는 데 있다. 이러한 조합형 프리미티브 중 다수는 시간이 지나며 발견되겠지만, 몇 가지 예시는 다음과 같다.

1. **허가 불필요한 시장(Permissionless Markets)**: 오늘날 대부분의 시장은 기능하기 위해 특정 주체(예: 거래가 발생하는 플랫폼 제공자)의 허가가 필요하다. 그러나 여러 저수준 프리미티브를 결합하면 이러한 통제(intermediation)를 제거(disintermediate)하고, 과거에는 존재할 수 없던 영역에서도 허가 불필요한 시장이 발전할 수 있다. 이를 위해서는 다음이 필요하다.
 1. 거래 수단(medium of exchange) 및 계정 단위(unit of account) 역할을 하는 네이티브 통화
 1. 동적 교차통화 변환(dynamic cross-currency conversion)은 다중 통화 환경을 쉽게 만들지만, 사용자는 일반적으로 단일 기준 통화를 선호한다.
 2. 자산의 검증 가능한 소유권
 3. 자산의 개인 간/허가 불필요 전송(peer-to-peer, permissionless transfer)
 4. 검열 저항적(censorship-resistant) 마켓플레이스 애플리케이션(탐색성과 매칭 기능 제공)
2. **파생 자산(Derivative Assets)**: 검열 저항적 자산 저장소를 제공하는 것 자체도 의미 있지만, 여러 저수준 프리미티브를 결합함으로써 기존 자산, 트랜잭션, 논리를 결합해 무한히 다양한 새로운 자산을 창조할 수 있다. 이러한 자산은 리스크 관리 목적에 맞춰 맞춤화될 수 있다. 이를 위해서는 다음이 필요하다.
 1. 자산의 검증 가능한 소유권
 2. 자산의 프로그래밍 가능한 에스크로(programmatic escrow)
 3. 프로그래밍 가능한 권리 이전(programmatic rights transfer)

3. **오픈 상태 구성 요소(Open State Components):** 모든 애플리케이션은 (권한이 부여된 경우) 자산이나 사용자 관련 데이터를 포함한 공유 상태 풀(shared pool of state data)에 접근할 수 있다. 이를 통해 구성 요소들은 오늘날의 마이크로서비스(microservices)처럼 특정 기능을 수행하고, 이를 조합해 더 큰 비즈니스 목표를 달성할 수 있다. 공개적이기 때문에 경쟁이 자연스럽게 작동하며, 가장 우수한 구성 요소가 선택되어 사용된다. 이를 위해서는 다음이 필요하다.

1. 공유 데이터 풀(shared data pool)
2. 사용자 주권 데이터(user-sovereign data)
3. 검증 가능한 프로세스 무결성(verifiable process integrity)

미래(The Future)

미래는 인터넷처럼 무한히 구성 가능하다. 그것이 정확히 어떤 형태를 띠지는 모르지만, 그 방향을 결정할 핵심 요인들과 이를 실현할 주요 도구들은 예측할 수 있다. **프라이버시(Privacy):** 기본적으로 블록체인 상의 활동과 데이터는 공개된다. 그러나 프라이버시의 본질은 **선택(choice)**에 있다 — 자신의 활동을 공개할지, 혹은 숨길지를 스스로 결정하는 것이다. 기본 기술 도구가 이러한 프라이버시 보호 기능을 제공하지는 않지만, 그 위에 구축된 다양한 솔루션을 통해 이를 구현할 수 있다. 약한 형태로는 데이터를 체인에 기록하기 전에 암호화할 수 있다. 이는 데이터 자체의 무결성을 보호하지만, 여전히 트랜잭션은 추적 가능하며 정교한 분석을 통해 실제 발생한 내용을 추론할 수 있다. 따라서 **영지식 증명(Zero-Knowledge Proofs)**과 같은 새로운 기술은 데이터 자체뿐 아니라 이를 수정하는 연산(computations)까지도 비공개로 유지할 수 있는 가능성을 제시한다.

이 기술은 NEAR 플랫폼에 처음부터 내장되어 있지는 않지만, 커뮤니티가 이를 추진한다면 구현이 가능하다. [<https://near.org/papers/the-official-near-white-paper/>]

프라이빗 샤드(Private Shards): 모든 블록체인 사용 사례가 각 트랜잭션마다 공용 체인의 완전한 보안과 보호를 요구하는 것은 아니다. 때로는 사용자 컨소시엄이나 단일 주체가 검증 과정을 직접 통제하고, 주기적으로 메인 체인에 체크포인트를 기록하여 보안, 검증 또는 활동 내역 동기화를 수행하는 독립 체인을 운영하기를 원할 수 있다. 이러한 경우, 특정 샤드는 미리 정의된 검증자 세트를 사용하도록 구성되어 “프라이빗”하게 운영될 수 있다.

모바일 노드(Mobile Nodes): 샤딩은 네트워크의 총 처리 능력이 연결된 CPU 수에 비례하는 **수평 확장(horizontal scaling)** 기술이다. 즉, 검증 과정에 참여하여 커뮤니티 클라우드를 지원하는 장치가 많을수록 네트워크가 처리할 수 있는 트랜잭션 처리량도 증가한다. 전 세계에는 작동 가능한 CPU를 갖춘 수십억 개의 장치가 존재하며, 이 중 일부만 참여하더라도 네트워크는 탁월한 확장성을 달성할 수 있다. 추가 용량이 필요할 경우, 노드를 실행하기 위한 요구사항을 조정해 모바일 기기조차 참여할 수 있도록 만드는 것도 가능하다. 엔지니어링 트레이드오프는 존재하지만, 이는 수십억 개의 스마트폰이 네트워크의 일부로 작동할 수 있는 잠재력을 의미하며, 미래 연구에 매우 흥미로운 영역이다.

사물인터넷(IoT, Internet of Things): IoT 장치는 모바일 기기보다 더욱 특수한 사례로, 처리 능력은 낮지만 사용 가능한 CPU의 수는 훨씬 많다는 점에서 독특하다.

조합형 구성 요소(Composable Components, 오픈 웹 The Open Web): 글로벌 자유 통화로 시작하여 “죽지 않는(unkillable)” 애플리케이션으로 이어지는 오픈 웹의 꿈은, 결국 모든 애플리케이션이 손쉽게 조합되어 새로운 기능을 만들어내는 세계를 의미한다. 하드웨어 관점에서 보면, 스마트폰의 GPS, 카메라, 인터넷 연결이 열어준 가능성을 떠올리고, 여기에 소프트웨어의 유연성을 결합한 형태다. 이러한 세상에서는 혁신이 얼마나 빠르게 일어날지 예측할 수 없다. NEAR의 글로벌 상태(global state)는 모든 애플리케이션이 접근 가능하므로, 이러한 미래는 현실이 될 것이다.

다음 단계(What's Next?)

첫걸음을 내딛자! <https://near.org/newsletter>에서 뉴스레터를 구독하고 NEAR의 최신 소식을 받아볼 수 있다. 프로토콜 개발은 오픈소스로 진행되고 있으며 <https://github.com/near>에서 코드를 확인하고, <https://docs.near.org>에서 코드 작동 방식과 구축 예시를 배울 수 있다. 또한 <https://near.chat> 또는 <https://stackoverflow.com/questions/tagged/nearprotocol> (<https://stackoverflow.com/questions/tagged/nearprotocol>)에서 질문할 수도 있다.

경험 수준이나 기술 숙련도와 관계없이 누구나 참여할 수 있다. 지금 NEAR의 여정에 함께하고 미래를 만들어가자.