

바운드리스 프로토콜

리스크 제로 팀(The RISC Zero Team)

요약(Abstract)

바운드리스(Boundless)는 모든 블록체인이 ZK 기반으로 검증이 가능한 연산 능력을 확장하기 위해 설계된 탈중앙화 시장, 집계 및 스마트 컨트랙트 결제 프로토콜이다. 바운드리스 프로토콜에서는 GPU 및 기타 일반 하드웨어를 운영하는 "증명자(provers)"들이 효율적이고 개방된 시장을 통해 온체인 또는 오프체인에서 "요청자(requestors)"가 제출한 영지식 증명(ZK Proof) 요청을 처리하기 위해 경쟁한다. 요청은 "스팟(spot)" 시장에서 발생할 수도 있고, "서비스 계약(service agreement)" 형태로 이루어질 수도 있으며, 두 경우 모두 결제는 요청자가 지정한 목적 체인(destination chain)에서 증명의 검증과 동시에 처리된다.

주요 기여점은 다음과 같다. 첫째, 증명의 복잡도와 상대적 비용을 측정하여 사기 및 스팸을 방지하는 메커니즘인 "검증 가능한 작업 증명(Proof of Verifiable Work)"을 통해, 증명자가 유용한 "ZK 마이닝" 활동에 참여하도록 토큰 보상을 제공함으로써 인센티브를 부여한다. 둘째, "연산의 금융화(Financialization of Compute)"를 통해 암호경제학적 인센티브를 활용하여 증명자와 요청자를 신뢰할 수 있고 자본 효율적인 방식으로 연결하는 문제를 해결한다. 마지막으로, 바운드리스는 이러한 개념들을 실질적으로 구현한 실용적인 시스템이다.

목차(Contents)

1 서론	2
1.1 ZK 채택의 여정(The road to ZK adoption)	3
1.2 남아있는 마지막 과제(The last remaining challenge)	4
2 바운드리스(Boundless)	4
2.1 연산의 금융화(The financialization of compute)	5
2.1.1 공급이 충분한 리소스 자원(An abundant commodity)	5
2.1.2 무신뢰 거래(Trustless trading)	5
2.2 검증 가능한 연산을 위한 스팟 시장(A spot market for verifiable compute)	6
2.2.1 이행 보장(Fulfillment guarantees)	8
2.2.2 탈중앙화(Decentralization)	9
2.3 토큰(A token)	9
2.3.1 검증 가능한 작업 증명(Proof of verifiable work)	10
2.3.2 볼트(The Vault)	11
2.4 검증 가능한 서비스 계약(Verifiable service agreements)	11
2.4.1 데이터 가용성에 관한 유의사항(A note on data availability)	13
3 구현(Implementation)	13
4 결론(Conclusion)	13

1 서론(Introduction)

연산 자원을 거래할 수 있다는 개념은 컴퓨팅 산업의 역사만큼이나 오래된 것이다. 지난 수십 년 동안 다양한 거래 모델이 등장했으며, 그 결과 오늘날 클라우드 컴퓨팅과 온체인 실행(on-chain execution) 같은 현대적 접근 방식으로 발전했다.

그러나 클라우드 컴퓨팅이 뛰어난 규모의 경제를 누리는 것과 달리, 온체인 실행 방식에서는 이러한 경제적 이점을 찾아볼 수 없다.

이러한 차이가 발생하는 원인은 바로 신뢰(trust)의 문제에서 기인한다. 클라우드 서비스의 사용자들은 서비스 공급자가 연산을 성실하게 수행할 것으로 암묵적으로 신뢰하는 반면, 블록체인의 사용자들은 특정 노드를 개별적으로 신뢰하지 않는다. 대신 블록체인 프로토콜 자체를 신뢰하며, 이 프로토콜에는 부정직한 노드를 방지하는 메커니즘이 내장되어 있다. 그러나 이 메커니즘(인센티브를 받고 의심스러운 연산을 재실행하는 검증자 메커니즘)은 온체인에서 수행 가능한 연산량에 본질적인 한계를 부과한다. 온체인 연산에 대한 수요가 이러한 한계를 초과하기 때문에 블록체인은 요청받은 연산 중 어느 것을 처리할지 결정할 수 있는 추가적인 메커니즘을 필요로 하게 된다.

이더리움(Ethereum)과 같은 체인에서는 이를 ‘가스 시장(gas market)’이라는 방식으로 해결하며, 이 시장은 제한된 온체인 연산 자원을 경매를 통해 할당한다. 가스 시장은 여러 가지 바람직한 특성을 가지지만, 근본적인 문제인 ‘온체인 연산 자원의 희소성’을 해결하지는 못하고 있다. 결국 가장 높은 마진을 가진 연산 요청만 처리되고 나머지 요청들은 비용 경쟁에서 밀려나며, 결과적으로 하나의 작업을 위해 다른 작업을 희생하는 구조가 되어버린다. 이러한 문제를 해결할 수 있는 보다 발전된 접근 방식으로 영지식 증명(ZKPs, Zero Knowledge Proofs)이 등장하였다. 영지식 증명은 노드가 특정 연산을 수행하여 특정 결과를 얻었다는 사실을 증명할 수 있도록 함으로써 신뢰 구조 자체를 근본적으로 변화시킨다. 이를 통해 사용자는 연산을 수행한 노드를 신뢰하지 않더라도 연산의 결과를 신뢰할 수 있게 된다. 이와 같은 방식으로 신뢰 구조를 변화시키는 영지식 증명 기술은 기존의 의심 기반의 재실행 필요성을 없애며, 확장 가능(scalable)하고, 신뢰가 필요 없으며(trustless), 탈중앙화(decentralized)된 연산 환경으로 나아가는 길을 열어준다.

1.1 ZK 채택의 여정(The road to ZK adoption)

앞서 언급한 논리와 개념은 이미 오래전부터 알려져 있었지만, 실제로 ZK 기술의 현실적인 채택은 최근까지 매우 제한적이었다.

영지식 증명(ZK)의 초창기 역사는 특수한 문제만 해결 가능한 개별적이고 제한적인 기법들의 집합체였다. 당시의 대표적인 예로는 RSA나 ECDSA 같은 공개키 디지털 서명(public key digital signature) 기법이 있다.

시간이 흐르면서 계산 복잡도 이론(computational complexity theory), 특히 확률적 검증 가능 증명(probabilistically checkable proofs, PCP)의 발전이 이루어졌으며, 이를 통해 ZK가 범용적인 연산(general-purpose computation)을 처리할 수 있을 것이라는 강력한 이론적 근거가 마련되었다. 이어서 실용적인 알고리즘이 등장했고, 그 성과는 튜링상(Turing Award) 수상으로도 이어졌다.

하지만 이러한 알고리즘들은 실용적이긴 했지만 사용하기 쉽다고는 볼 수 없었다. 그 본질적인 특성상 응용 암호학(applied cryptography)에 대한 심도 있는 전문성과 최신 이론적 성과에 대한 높은 이해를 갖춘 소수 전문가만 접근할 수 있었다. 이러한 상황에서도 2016년에 Zcash가 출시되면서 ZK 기술의 현실적인 구현 가능성이 입증되었다. 이후 ZK 기술을 더 효율적이고 다목적이며 안전하고 쉽게 활용할 수 있도록 하기 위한 연구가 폭발적으로 증가했고, 그 결과 Bulletproofs(2018), Circom(2019), Halo(2019) 등 혁신적인 기술들이 등장하였다. 같은 시기 STARKs의 발전과 2020년에 출시된 CairoVM은 신뢰 설정(trusted setup) 없이도 안전하고 프로그래밍 가능한 ZK 환경이 가능하다는 사실을 보여주었다.

2022년, RISC Zero는 최초의 범용적인 표준 아키텍처(RISC-V) 기반의 영지식 가상 머신인 zkVM을 공개했다. zkVM은 산업에 획기적인 혁신을 가져왔다. 과거 ZK 기술을 도입하려면 특수한 프로그래밍 언어나 회로 언어(circuit languages)를 사용해야만 했던 것과 달리, zkVM은 RISC-V를 지원하는 모든 프로그래밍 언어를 활용할 수 있도록 만들었다. 대표적으로 Rust 언어가 이에 해당하며, Rust는 뛰어난 개발 생태계와 풍부한 라이브러리를 갖추고 있어 블록체인 분야를 비롯한 다양한 산업에서 폭넓게 채택되고 있다.

이듬해 RISC Zero는 최초의 "타입 1(Type 1)" zkEVM인 Zeth를 공개하면서 zkVM의 강력한 성능을 입증하였다. Zeth의 등장은 업계에 큰 반향을 불러일으켰다. 이전까지 zkEVM 개발은 수년이 걸릴 것으로 인식되었으나, Zeth는

단 3명의 개발자로 구성된 소규모 팀이 한 달도 채 걸리지 않고 이를 완성할 수 있음을 증명했기 때문이다. 이러한 파괴적인 혁신과 함께 zkVM이 처음 출시된 이후 누적된 성능 개선 효과는 업계 전반의 프로젝트 로드맵을 근본적으로 변화시키기에 이르렀다. 이제 ZK 기술은 단순히 "가능성"의 영역을 넘어, zkVM을 통해 빠르고 손쉽게 도입할 수 있는 기술로 자리매김하고 있다.

1.2 남아있는 마지막 과제(The last remaining challenge)

그러나 ZK 기술을 실제로 채택하는 것은 단순히 Rust로 코드를 작성한 후 zkVM에서 실행하는 것처럼 간단하지 않다. 실제적 의미의 ZK 기술 채택을 위해서는 여전히 남아있는 마지막 과제인 엔드 투 엔드(end-to-end) 통합 문제와 마주해야만 한다.

이러한 과제의 본질적 난점은 온체인 연산을 위한 가스 시장(gas market)이 블록체인 프로토콜 자체에 내장되어 있는 반면, ZK 증명(ZK proving)에 접근하는 메커니즘은 그렇지 않다는 점에서 비롯된다. 따라서 ZK 기술을 도입하려는 팀들은 필연적으로 증명을 처리할 자체적인 파이프라인을 구축해야 한다. 이는 결코 간단한 과제가 아니다. 적절한 탈중앙화 인프라의 부족, 기존 온체인 실행과 ZK 기반 실행 간의 제어 흐름(control flow) 차이, 인센티브 관리(incentive management), 그리고 증명 능력(proving capacity)을 초기화하고 지속적으로 유지하는 문제 등 복합적이고 다양한 어려움을 수반한다.

zkVM이 ZK 애플리케이션의 작성 방식을 획기적으로 바꾼 것처럼, 이제는 이러한 애플리케이션을 배포하고 통합하는 방식에도 근본적인 혁신이 필요하다.

2 바운드리스(Boundless)

ZK 기술 채택 과정에서 직면하게 되는 다양한 문제를 해결하기 위해 스마트 컨트랙트와 새로운 암호학적 원시 요소(cryptographic primitives), 오프체인 인프라(off-chain infrastructure)를 결합하여 완전한 탈중앙화를 구현한 ZK 기반 프로토콜 바운드리스(Boundless)가 제안되었다. 바운드리스는 연산의 금융화(Financialization of Compute)를 통해 모든 프로토콜이나 블록체인이 허가가 필요 없는 시장(permissionless marketplace)에서 안전하고 효율적이며 신뢰 가능한 방식으로 연산을 수행할 수 있도록 지원한다.

- 바운드리스의 핵심은 호환 가능한 모든 블록체인에 배포된 스마트 컨트랙트를 통해 구현된다. 각각의 배포는 독립적이며, 탈중앙성(decentralization), 실시간성(liveness), 크립토 이코노미 기반의 보안성(cryptoeconomic security) 등 배포된 체인의 속성을 그대로 상속받는다.
- 바운드리스는 인프라 제공자들이 여러 블록체인 배포 환경에 동시에 원활히 참여할 수 있도록 해, 모든 체인에 충분한 증명 유동성(proving liquidity)을 확보한다.
- 영지식 증명(zero knowledge proofs)과 검증 가능한 작업 증명(proof of verifiable work)이 지닌 보안 속성을 활용하여 신뢰가 필요 없는 방식으로 연산에 대한 결제를 교환할 수 있도록 지원한다.

- 바운드리스는 시장이다. 즉, ZK 기반의 검증 가능한 연산(verifiable compute)을 상품으로 취급하여, 이를 직접적으로(스팟 시장에서) 또는 간접적으로(서비스 계약을 통해) 거래 가능하도록 함으로써, 가격 발견(price discovery), 실시간성(liveness), 검열 저항성(censorship resistance)과 같은 문제를 해결한다.
- 바운드리스의 연산 능력은 증명자들이 운영하는 연산 자원과 선형적으로 확장된다. 즉 하드웨어를 두 배 늘리면, 연산 처리 능력도 두 배로 증가한다.

2.1 연산의 금융화(The financialization of compute)

본 섹션에서는 바운드리스 프로토콜의 근본적인 개념을 소개한다. 첫째, 검증 가능한 연산(verifiable compute)을 상품(commodity)으로 간주할 수 있다는 점이며, 둘째로 검증 가능한 작업 증명(proof of verifiable work)을 통해 이러한 상품을 신뢰할 수 있게 측정하고 거래할 수 있도록 만든다는 점이다. 바운드리스는 이러한 특성을 기반으로 시장의 원리를 적극적으로 활용해 허가가 필요 없고(permissionless), 신뢰가 필요 없는(trustless) 방식으로 검증 가능한 연산 능력을 거래하고 활용할 수 있게 해준다. 결과적으로 앞서 제시된 엔드 투 엔드(end-to-end) 통합의 복잡한 과제를 효율적이고 신뢰할 수 있는 방식으로 해결한다.

2.1.1 공급이 풍부한 리소스 자원(An abundant commodity)

ZKP 기술은 이제 MacBook Pro나 게이밍 PC와 같은 일반 소비자용 하드웨어에서도 상당히 복잡한 수준의 영지식 증명(ZKP)을 생성할 수 있을 만큼 성숙한 단계에 도달했다. 더욱이, 이러한 증명을 생성하는 데 필요한 소프트웨어 역시 무료이며 오픈소스로 제공된다. 이처럼 낮은 진입 장벽 덕분에 검증 가능한 연산 능력은 일반적으로 공급이 풍부한 리소스 자원(abundant commodity)과 같은 특성을 나타낼 것으로 기대된다.

이러한 특성은 사용자와 프로토콜에게는 안정적인 가격으로 쉽고 신뢰성 있게 검증 가능한 연산 능력에 접근할 수 있다는 점을 의미한다.

업적 규모로 증명 서비스를 제공하는 증명자(provers)의 관점에서는 자원 관리(resource management), 캐파 계획(capacity planning), 가격 책정(pricing), 서비스의 신뢰성(reliability) 향상 등을 통해 부가가치를 창출할 수 있다. 또한, 다양한 서비스 계약(service agreements)이나 차별화된 상품을 판매하고, 개인 사용자나 아마추어 증명자들로 구성된 풀(pool)을 관리하거나 조율하여 추가적인 부가가치를 더할 수도 있다.

마지막으로 거래자(traders)의 경우 가격 발견(price discovery)에 기여하거나, 시장 참여자들이 가격 변동성에 대비할 수 있는 금융 상품을 제공함으로써 부가가치를 창출할 수 있다(자세한 내용은 이후의 서비스 계약 항목 참조).

2.1.2 무신뢰 거래(Trustless trading)

모든 분쟁이 동시에 관찰 가능한 상호주관적 진실(intersubjective truth)에 근거하여 해결되는 물리적 상품 거래와 달리, ZK 기반의 검증 가능한 연산과 관련된 모든 정보는 제한된 불확실성(bounded uncertainty, 즉 표준 암호학적 가정 하의 확률적 건전성) 내의 객관적 진실(objective truth)의 영역에 속한다. 적절하게 구현된 회로 및 소프트웨어와 적합한 암호학적 매개변수를 사용할 경우, 이는 사실상 객관적 진실과 동등하게 여겨질 수 있으며,

이러한 특성으로 인해 ZK 기반의 검증 가능한 연산 능력은 신뢰가 필요 없는(trustless) 환경에서 매우 유리하게 활용될 수 있다.

하지만 이러한 특성만으로는 검증 가능한 연산 능력을 효율적으로 거래할 수 없다. 효율적인 거래를 위해서는 특정 작업에 소모된 연산의 양을 신뢰할 수 있고 안전하게 측정할 수 있는 방법이 필요하다. 바운드리스 프로토콜에서는 이를 "검증 가능한 작업 증명(proof of verifiable work)"이라는 간단하지만 견고한 메커니즘으로 구현하고 있다. 이 메커니즘을 통해 증명자(prover)는 특정 증명을 생성하기 위해 수행된 연산량을 투명하게 공개할 수 있다. 결과의 객관성(objectivity)과 검증 가능한 작업 증명을 통해 보고된 측정값(metrics)의 무결성(integrity)이 결합되었을 때, 비로소 검증 가능한 연산 능력의 신뢰가 필요 없는(trustless) 거래가 실현 가능해진다.

2.2 검증 가능한 연산을 위한 스팟 시장(A spot market for verifiable compute)

간단한 예시로 시작해 보자. Alice는 온체인 앱을 이용하기 위해 영지식 증명(ZKP)을 제출해야 한다. 이 ZKP는 충분한 연산 자원과 필요한 데이터에 접근 가능한 사람이라면 누구든지 생성할 수 있다. Alice는 ZKP를 스스로 생성하고 싶지 않아서 대신 이를 생성하고 제출해 줄 사람을 찾기 시작한다. Bob과 Charlie는 둘 다 Alice가 비용과 합리적인 이윤을 제공할 의사가 있다면 ZKP를 생성해줄 용의가 있다.

Alice는 필요한 연산 작업을 기술할 수 있고, 이 설명을 바탕으로 Bob과 Charlie는 해당 요청을 수행하는 데 필요한 작업량을 추정할 수 있다. 따라서 시장의 관점에서 남은 유일한 질문은 누가 가장 낮은 가격으로 이를 수행할 수 있는지는 것이다.

바운드리스 마켓플레이스는 이 질문에 대해 역 더치 경매(reverse Dutch auction) 방식을 활용하여 답을 제시한다. 역 더치 경매에서 가격 발견은 다음과 같이 이루어진다:

1. 요청자(Alice)는 공개된 공간(바운드리스 마켓플레이스)에서 온체인 또는 오프체인으로 자신의 요청을 공개한다.
2. 처음에 요청자는 요청이 바로 처리되는 것에 대해 높은 비용을 지불할 의사가 없다. 요청자는 낮은 가격으로 요청을 처리할 사람이 있는지 일정 시간이 지날 때까지 기다린다.
3. 시간이 지남에 따라 요청이 처리되지 않으면, 요청자는 자신이 지불할 의사가 있는 가격을 점진적으로 높인다.
4. 결국 다음 중 하나가 발생한다:
(a) 가격이 올라서 누군가(Bob, Charlie, 또는 다른 사람)가 요청을 처리할 만큼 충분히 높아지거나, (b) 가격이 너무 높아져 요청자가 더 이상 가격을 올릴 의사가 없어지는 상황이 발생한다. 여전히 아무도 요청을 처리하지 않으면 해당 요청은 미처리 상태로 남게 된다.

이러한 방식은 여러 가지 이점을 제공한다:

- 더치 경매는 첫 번째 입찰이 확정되자마자 종료된다. 이 속성으로 인해 더치 경매는 지연시간이 짧아야 하는 상황이나 온체인 구현에 특히 적합하다.

- 역 경매(reverse) 방식은 일반적인 경매 방식보다 프로토콜을 간소화한다. 일반적인 경매 방식에서는 증명자가 자신들의 가격과 용량을 공개해야 하는데, 신뢰 없이 이를 수행하는 것은 쉽지 않다. 역 경매는 이런 문제를 피할 수 있다.
- 역 더치 경매는 효율성과 안정성 측면에서 바람직한 특성을 가진다. 결정적으로, Alice는 가장 낮은 가격으로 ZKP를 확보할 수 있으며 추가적인 비용도 최소화된다.

따라서 Alice가 ZKP가 필요할 때, 그녀는 다음과 같은 데이터를 포함한 요청을 바운드리스 마켓플레이스에 공개한다:

- 필요로 하는 ZKP에 대해 기계가 판독 가능할 수 있도록 설명 (해당 요청을 수행하는 데 필요한 추가 메타데이터 포함)
- 경매 시작 시점을 나타내는 타임스탬프
- 단조 증가(monotone, non-decreasing) 형태의 가격 함수(ETH 또는 ERC20 기반으로 표시됨)
- 요청을 고정(lock)하고자 하는 증명자가 스테이킹할 수 있는 옵션 형태의 보증금(bond)
- 만료일 또는 마감 시한(deadline)

이러한 요청의 매개변수(parameter)를 설정한 후, Alice는 요청을 게시하는 데 다음과 같은 몇 가지 옵션을 선택할 수 있다:

- 선택한 체인에 있는 바운드리스 마켓플레이스 스마트 컨트랙트에 온체인으로 요청을 게시하거나,
- 요청을 서명하여 오프체인 상에서 하나 이상의 별도의 프로토콜(제3자 가십 프로토콜, 거래소, 리스팅 서비스 등)에 게시할 수 있다.

어떤 경우이든, Alice의 요청은 궁극적으로 Alice가 지정한 체인의 바운드리스 마켓플레이스 컨트랙트를 통해 완료된다.

Alice의 요청이 공개된 이후, Bob과 Charlie는 모두 요청을 수행하는 데 필요한 작업량을 평가하기 시작한다. 이들은 요청에 포함된 설명(각주 3 참조), 요청의 만료일 또는 마감시한, 자신들의 현재 작업량, 시장에서 이용 가능한 다른 기회들, 원하는 이윤율, 다른 경쟁자가 더 낮은 가격으로 수행할 가능성 등을 고려해 평가를 진행한다. 이러한 요소들을 바탕으로 두 사람은 요청을 수행할 의향이 있는 최소한의 가격을 각각 정한다. 편의상, Charlie가 Bob보다 낮은 가격을 선택했다고 가정하자.

시간이 지나도 요청이 수행되지 않으면, 요청 가격은 가격 함수에 따라 프로그래밍적으로 증가한다. 결국 가격이 Charlie가 정한 기준점(threshold)에 도달하게 된다. 이 시점에서 Charlie는 다음 두 가지 선택지를 갖게 된다:

1. 요청을 즉시 수행한다.

Charlie는 경쟁자가 매력을 느끼고 참여할 만큼 가격이 올라가기 전에 최대한 빨리 증명을 생성하고 요청을 처리하여 보상을 얻을 수 있다.

2. 요청을 즉시 고정(lock)한다.

이 조치를 취하면 Charlie는 해당 요청을 독점적으로 수행할 수 있는 권한을 얻게 된다. 요청을 고정하면 다른 경쟁자가 마지막 순간에 요청을 가로챌 위험 없이 증명 생성 작업을 진행할 수 있다. 이 독점적 권한을 확보하려면 Charlie는 일정량의 자금을 스테이킹(stake)해야 하며, 만약 정해진 마감기한 내에 요청을 이행하지 못할 경우 이 스테이킹된 자금을 잃게 된다.

어떤 상황에서는 Alice가 요청의 고정을 허용하지 않는 것을 선호할 수도 있다. Alice는 이 내용을 요청에 명시적으로 표시할 수 있으며, 이 경우 위의 (2)번 옵션이 금지된다.

Charlie가 요청을 완료했을 때, Charlie는 프로그래밍적으로 증명자 수수료(prover's fee)를 지급받게 된다. 이 수수료는 요청의 최종 가격에서 시장의 수수료(시장 점유율에 따라 결정되며, 마켓플레이스의 볼트(vault)에 누적됨)를 제외한 금액이다. 요청의 최종 가격은 Charlie가 요청을 사전에 고정했는지 여부에 따라, 고정 시점의 가격 또는 실제 수행 시점의 가격 함수에 의해 결정된다. 증명 검증에 소모되는 가스 비용을 절약하기 위해(그리고 Alice의 요청이 허용한 경우), Charlie는 증명을 다른 증명들과 함께 일괄 처리(aggregated batch) 형태로 제출할 수도 있다.

2.2.1 이행 보장(Fulfillment guarantees)

모든 것이 계획대로 진행된다면 Alice는 지정된 마감 기한 전에 가능한 가장 낮은 가격으로 증명을 받게 된다. 그러나 만약 누군가가 그녀의 요청을 고정(lock)했지만 (의도적이든 아니든) 마감 기한을 지키지 못한다면 어떻게 될까?

Charlie가 Alice의 요청을 고정했으나 마감 기한을 놓쳤다면, Charlie는 자신이 스테이킹한 자금을 잃게 되고 Alice는 환불을 받게 된다. 이후의 과정은 Alice가 처음 요청을 어떻게 설정했는지에 따라 달라진다:

- 가장 간단한 경우, 잃게 된 스테이킹 자금은 마켓 컨트랙트로 보내지고 (이 자금은 포인트 메커니즘을 통해 토큰 보유자에게 분배될 수 있다), 해당 요청은 처리 완료된 것으로 간주된다.
- 또는, 요청이 허용할 경우 해당 요청은 "재시도(retry)" 단계로 넘어간다. 이 재시도 단계에서는 요청 고정(locking)이 비활성화되며, 증명자들은 기존 증명자가 잃은 스테이킹 금액의 일부를 보상으로 받는다(나머지는 마켓 컨트랙트로 보내진다). Alice가 스테이킹 요구 조건을 적절히 설정했다면, 증명자들은 이 프리미엄 보상을 인식하고 Alice의 요청을 처리하기 위해 경쟁적으로 참여하게 된다.

Alice에게는 다른 전략들도 존재한다. 예를 들어, Alice가 지금 당장 증명이 필요하다면, 매우 짧은 마감 기한과 요청 고정 불가 설정, 그리고 높고 고정된 가격을 가진 요청을 제출할 수 있다. 이렇게 하면 Bob과 Charlie가 이 요청을 보고 높은 프리미엄을 얻기 위해 경쟁적으로 빠르게 요청을 처리할 것이다.

또 다른 경우로, Alice가 3일 내에 요청 처리를 원한다고 가정해 보자(이는 실제 증명을 생성하는 데 필요한 시간보다 훨씬 긴 기간이다). 이 경우 Alice는 보다 짧은 마감 기한(예: 12시간), 합리적인 초기 가격, 매력적인 최대 가격, 그리고 적당히 높은 수준의 스테이킹 요구 조건을 설정한 요청을 제출할 수 있다. 만약 이러한 요청이 아무런 관심을 끌지 못하면, 그녀는 아직 충분한 시간이 남아 있으므로 더 높은 가격이나 다른 매개변수를 설정하여 다시 요청을 제출할 수 있다.

2.2.2 탈중앙화(Decentralization)

대규모 증명(proof) 생성은 산업적 프로세스로서 다른 산업적 프로세스들과 마찬가지로 규모의 경제(economies of scale)를 갖는다. 동일한 조건이라면, 강 옆에 위치한 데이터 센터는 여러 대의 가정용 게이밍 PC로 이루어진 풀(pool)보다 훨씬 낮은 비용으로 더 많은 증명을 생성할 수 있을 것이다. 여기에서 우리는 다음과 같은 중요한 질문을 던질 수 있다: 무엇이 독점(monopoly)의 형성을 막을 수 있을까?

이 질문에 답하기 위해 이전 예시에서 등장한 증명자인 Bob과 Charlie의 사례를 다시 살펴보자. 가령, Charlie가 경쟁자들보다 훨씬 큰 규모로 성장하여 증명 시장에서 대부분의 요청을 처리하게 되었고, 반면 Bob은 요청을 거의 고정(lock)하지 못해 결국 시장에서 밀려났다고 가정해 보자.

Charlie가 합리적인 가격으로 요청을 계속 처리한다면 시장에 큰 문제가 되지 않는다. 그러나 Charlie가 자신의 독점적 지위를 이용하여 가격을 높이거나 특정 유형의 요청을 거부하기 시작한다면 어떻게 될까?

Charlie가 자신의 독점적 지위를 남용하기 시작하면, Bob에게는 새로운 기회가 열리게 된다. Charlie가 비록 증명 시장을 장악할 수는 있지만, 증명 시장에 참여하는 데 필요한 하드웨어까지 독점할 수는 없기 때문이다. 실제로 Bob은 이미 증명에 필요한 하드웨어(MacBook Pro, 게이밍 PC 등)를 보유하고 있으며, 수익성이 충분하다면 클라우드 인스턴스를 여러 대 추가로 활용하여 신속히 시장에 진입할 수 있다. 또한 바운드리스 소프트웨어 자체가 쉽게 배포 및 실행할 수 있는 구조로 설계되어 있으므로, Bob은 Charlie보다 저렴한 가격으로 서비스를 제공하거나 Charlie가 무시하는 요청들을 처리함으로써 수익을 창출할 수 있는 환경이 갖추어져 있다.

바운드리스 마켓플레이스의 목표는 사용자들에게 낮은 비용으로 신뢰할 수 있는 서비스를 제공하는 것이다. 이러한 목표는 Bob과 같은 개인이 이미 보유한 하드웨어를 사용하여 쉽게 시장에 진입함으로써 시장의 비정상적 상황을 빠르게 바로잡을 수 있도록 하는 구조적 설계를 통해 달성된다. 즉, 시장 원리(market forces)와 허가가 필요 없는 체인(permissionless chains)에 대한 의존이 바운드리스가 지향하는 탈중앙화의 핵심이다. 증명자로서 시장에 진입하는 데 있어 진입 장벽은 매우 낮다. 바운드리스 소프트웨어는 오픈소스로 제공되며 일반적으로 사용 가능한 하드웨어에서 쉽게 작동하기 때문에, 증명 공급(proving supply)은 특정 주체의 허가가 필요 없고 누구에게도 소유되지 않는다. 이상적으로는 이러한 방식으로 시장의 왜곡(aberration)을 즉각적으로 교정할 수 있을 만큼 충분한 "연산 유동성(compute liquidity)"이 확보될 것이다. 결국 바운드리스 마켓플레이스에 제기된 요청은 해당 요청을 수행하는 것이 최소한 한 명 이상의 참여자에게 경제적으로 이득이 된다면 언제나 공정한 가격으로 처리될 것이다.

2.3 토큰(A token)

바운드리스 마켓플레이스는 신뢰성 있고 원활한(low-friction) 거래와 무신뢰(trustless) 거버넌스를 촉진하기 위해 다음과 같은 특성을 지닌 새로운 토큰인 \$ZKC를 도입한다:

- **총 공급량 제한(Capped Supply)**

총 토큰 공급량에 상한(cap)을 설정하여 희소성을 유지한다.

- **가변적인 토큰 발행 속도(Variable Minting Rate)**

신규 토큰 발행 속도는 최종적으로 온체인 거버넌스를 통해 토큰 보유자(token holders)들이 제어하며, 이 발행 속도는 총 공급량 상한에 도달하는 시기를 결정하게 된다. 신규로 발행된 토큰은 아래에서 자세히 설명할 검증 가능한 작업 증명(Proof of Verifiable Work) 보상 메커니즘의 일환으로 증명자(provers)들에게 분배된다.

- **시장 수수료 접근 포인트(Points Generation)**

토큰 보유자는 시장에서 발생하는 수수료에 접근할 수 있는 포인트(points)를 생성할 수 있다.

- **스테이킹 가능성(Staking Capability)**

보유한 토큰은 스팟 시장(spot market)이나 서비스 계약(service agreements)에 스테이킹할 수 있다.

- **가변적 보상 주기(Variable Reward Frequency)**

토큰과 포인트 보상의 분배 주기는 고정되어 있지 않으며, 온체인 거버넌스를 통해 결정된다.

이러한 개념에 대해서는 아래 섹션에서 더욱 구체적으로 설명한다.

2.3.1 검증 가능한 작업 증명(Proof of verifiable work)

요청이 처리되면, 바운드리스 마켓플레이스는 보상 분배와 관련된 다양한 지표(metrics)를 자체적으로 추적한다. 구체적으로, 각 보상 주기(epoch)마다 바운드리스는 다음 두 가지 주요 항목을 추적한다.

- **누적 수수료(Cumulative Fees):** 시장에서 누적하여 수집된 수수료로, 시장에서 제공된 가치를 나타낸다.
- **누적 증명 사이클 수(Cumulative Number of Cycles Proven):** 시장에서 수행된 전체 작업량을 나타낸다.

또한 바운드리스는 이러한 지표를 증명자(prover)별로 세분화하여 추적하기 때문에, 특정 주기(epoch)에 특정 증명자가 처리한 요청과 관련된 수수료나 사이클의 비율을 쉽게 산출할 수 있다.

바운드리스의 스마트 컨트랙트는 위의 지표를 바탕으로 증명자들의 시장 기여도에 따라 신규로 발행된 토큰을 자동으로 보상으로 지급한다. 결과적으로 증명자는 두 가지 형태의 수익원을 얻을 수 있다. 첫째, 요청 처리 시 요청자로부터 지급받는 수수료(payment)이고, 둘째는 검증 가능한 작업 증명(Proof of Verifiable Work) 메커니즘을 통해 얻는 고유한 토큰 보상(native rewards)이다.

그러나 여기서 중요한 의문이 제기될 수 있다. 시장에 "기여(contribution)"한다는 것은 정확히 어떤 의미인가? 최종적으로 "기여"에 대한 정확한 정의는 토큰 보유자(token holders)의 거버넌스를 통해 결정되겠지만, 바운드리스 출시 단계에서는 다음 두 가지 핵심 지표가 사용된다.

- **제공된 가치(Value Delivered):** 시장에서 높은 비율의 수수료 수익 창출에 기여한 증명자는 그에 비례하여 높은 보상을 받는다.
- **수행된 작업량(Work Performed):** 전체 증명 사이클에서 높은 비율을 차지한 증명자는 역시 그에 비례하여 높은 보상을 받는다.

이 두 가지 지표는 다음 규칙을 통해 하나의 보상으로 통합된다. 즉, 각 증명자는 시장에서 자신이 창출한 수수료 비율과 자신이 수행한 작업 사이클 비율 중 더 작은 비율에 따라 보상을 받는다.

예를 들어, Bob이 시장에서 발생한 수수료의 25%를 창출했지만 실제 증명 사이클에서는 10%만 수행했다고 가정해 보자. 이 경우 그는 신규 발행된 토큰의 10%만 보상으로 받게 된다. 반대로 Charlie는 시장 수수료의 75%를 창출하고 전체 증명 사이클의 90%를 수행했다고 가정하면, 그는 신규 발행된 토큰의 75%를 보상으로 받게 된다.

위 예시에서 Bob과 Charlie가 모든 수수료와 사이클을 차지했지만, 실제로 신규 발행된 토큰은 $10\% + 75\% = 85\%$ 만 지급된다. 나머지 15%는 소각(burn)되어 결과적으로 토큰 발행률(minting rate)이 감소하게 된다.

증명된 작업 사이클 수를 추적하는 기능은 RISC Zero의 zkVM에서 제공하는 "검증 가능한 작업 증명(Proof of Verifiable Work)"이라는 고유한 메커니즘에 핵심적으로 의존한다. 이 기능은 증명자(prover)가 생성한 증명(proof)에 해당 증명을 만드는데 투입된 사이클 수에 대한 메타데이터(metadata)를 포함시킬 수 있도록 한다. 이 메타데이터는 암호학적으로 안전하게 보호되므로, 증명자가 증명의 유효성을 훼손하지 않고는 이를 조작하거나 변경할 수 없다. 또한 각 메타데이터는 고유한 불변의 값 nonce)을 포함하고 있어, 동일한 증명이 보상 메커니즘에서 중복하여 집계되는 것을 효과적으로 방지한다.

2.3.2 볼트(The Vault)

앞서 언급한 바와 같이, 바운드리스 마켓플레이스는 처리 완료된 모든 요청에 대해 일정 비율의 수수료를 징수한다. 이 비율을 테이크 레이트(take rate)라고 부르며, 토큰 보유자(token holders)들의 거버넌스를 통해 조정할 수 있다. 이렇게 시장에서 발생한 수수료는 궁극적으로 \$ZKC 보유자들에게 분배된다. 본 섹션에서는 이러한 수수료 분배 메커니즘을 자세히 설명한다.

또한 각 배포마다 볼트(vault)라 불리는 스마트 컨트랙트(변경 불가능하게 설계됨)가 존재하며, 여기에 \$ZKC 보유자들은 허가 없이(permissionlessly) 자신의 토큰을 잠글 수 있다(lock-up). 볼트에 잠긴 토큰은 양도 불가능한(non-transferable) 포인트(points)를 생성한다. 이 포인트는 추후 소각(burn)하여 수수료 풀에서 수수료를 추출하거나, 볼트에 잠긴 자신의 토큰을 해제(unlock)할 때 사용된다. 수수료 추출 및 토큰 해제는 비례 원칙(proportionality)에 따라 이루어진다. 예를 들어, 누군가 전체 미사용 포인트(outstanding points)의 1%를 소각하면, 소각 시점의 수수료 풀에서 1%의 수수료를 보상으로 받는다. 마찬가지로 본인이 보유한 포인트의 1%를 소각하면, 그 시점에 볼트에 잠긴 본인의 토큰 중 1%를 해제할 수 있는 권리를 얻게 된다.

포인트 생성 기능 외에도, 볼트에 잠긴 토큰은 서비스 계약(service agreements, 이후 섹션 참조)에 스테이킹하거나, 스팟 시장에서 요청을 고정(lock)하는 용도로도 활용할 수 있다. 볼트에 잠긴 모든 토큰은, 이와 같은 방식으로 스테이킹된 상태에서도 프로그래밍적으로 지속적으로 포인트를 생성하게 된다.

2.4 검증 가능한 서비스 계약(Verifiable service agreements)

검증 가능한 서비스 계약이란 증명자(prover)와 요청자(requestor) 간의 구속력 있는 약속으로, 일반적으로 요청자가 증명자에게 특정한 검증 가능한 작업을 수행하도록 강제할 수 있는 조건들을 명시한 계약이다.

서비스 계약은 구체적으로 바운드리스의 시장(market), 볼트(vault), 집계(aggregation), 전달(delivery) 및 보상(reward) 메커니즘을 활용하는 스마트 컨트랙트를 통해 구현된다. 바운드리스 프로토콜의 설계상 서비스 계약의 잠재적인 유형은 제한이 없다. 누구나 호환 가능한 스마트 컨트랙트를 배포하기만 하면 허가 없이(permissionlessly) 새로운 유형의 서비스 계약을 시장에 출시할 수 있다. 이러한 유연성 덕분에 다양한 사용 사례에 최적화된 맞춤형 계약을 만들 수 있으며, 동시에 바운드리스 마켓플레이스가 제공하는 규모의 경제(economies of scale)와 내재적 보상(intrinsic rewards)을 충분히 활용할 수 있다.

아래의 두 가지 예를 통해 서비스 계약의 구체적인 활용 사례를 살펴보자.

첫 번째 예시: Alice와 Charlie의 서비스 계약

최종 사용자(end user)인 Alice가 스팟 시장에서 증명자인 Charlie로부터 증명을 구매한 사례를 다시 떠올려 보자. Alice는 향후에도 더 많은 요청을 할 예정이므로 Charlie에게 별도의 제안을 한다. 즉, Alice는 소정의 초기 비용을 선불로 지불하고, 앞으로 1년에 걸쳐 제한된 빈도로 최대 100개의 유사한 요청을 미리 정해진 가격으로 구매할 수 있는 권리를 확보하고자 한다. 이들은 스마트 컨트랙트를 통해 계약 조건을 명시하여 서비스 계약을 공식화하며, Charlie가 계약 조건을 이행하지 않을 경우 Alice는 중개자 없이(disintermediated) 신뢰가 필요 없는(trustless) 방식으로 대응할 수 있는 장치를 마련하게 된다.

Charlie는 이 계약이 Alice와의 간편한 결제 경로(payment rail)를 확보할 수 있게 해주고, 장기적인 용량 계획(capacity planning)에도 도움을 주기 때문에 만족한다. Alice 또한 서비스 계약을 통해 증명의 스팟 시장 가격이 급등하는 상황에서 발생할 수 있는 비용 리스크를 줄일 수 있다. 한편, 예상했던 만큼의 요청을 실제로 하지 않더라도 Alice가 부담하는 비용은 초기의 계약 비용뿐이며, 미사용한 요청에 대해서는 추가 비용이 발생하지 않는다. 게다가 계약 조건이 허용한다면, Alice는 남은 서비스 계약을 다른 사용자에게 판매하여 일부 비용을 회수할 수 있다.

두 번째 예시: 프로토콜 개발자 Pam의 서비스 계약

서비스 계약을 원하는 사람은 Alice뿐만이 아니다. 이번에는 자신의 핵심 프로토콜을 안정적으로 운영하기 위해 충분한 증명 공급(proving supply)을 확보하려는 프로토콜 개발자 Pam의 사례를 살펴보자. Pam이 원하는 서비스 계약 조건은 다음과 같다.

- 향후 1년 동안 대량의 증명이 필요하며, 특히 갑작스러운 수요 급증이나 연말에 수요가 크게 증가할 가능성이 있다.
- 프로토콜이 원활하게 작동하도록 증명이 적시에 제공되어야 한다.

Pam은 이와 같은 조건을 충족할 수 있는 서비스 계약을 Charlie와 논의하지만, 이것만으로는 충분하지 않다. 과거 Charlie가 아무리 신뢰할 만했고 계약 뒤에 충분한 스테이킹을 걸었더라도, Charlie가 계약을 이행하지 못할 가능성은 항상 존재하기 때문이다. 이러한 상황이 발생하면 Pam은 언제든지 스팟 시장을 백업 옵션으로 사용할 수 있지만(섹션 2.2 및 2.2.2 참조), 업무량이 매우 많다는 점을 고려할 때 요청할 때마다 가격이 달라지는(on-demand) 방식을 피하고자 한다.

이러한 이유로 Pam은 Charlie뿐만 아니라 다른 증명자들과도 추가적인 서비스 계약을 맺어 여러 계약을 확보하고, 이를 하나의 바스켓(basket)에 모아 관리한다. 이러한 접근법은 신뢰성 문제를 근본적으로 해결한다. 바구니는

설계상 단일 실패 지점(single point of failure)이 없기 때문에 한 증명자가 계약을 이행하지 못하더라도 전체적인 증명 공급이 중단되지 않는다. 하지만 이렇게 모든 계약들의 총 용량은 예상 수요를 초과할 수 있으며, 필요량보다 상당히 클 수 있다. 따라서 Pam은 바구니를 분할(fractionalize)하여 초과 용량을 시장에 판매한다. 이를 통해 Pam은 비용 효율성을 높일 수 있으며, 구매자들 또한 일반적으로 대규모 프로토콜에서만 이용할 수 있는 고급 SLA(Service Level Agreement)를 합리적인 조건으로 이용할 수 있게 된다.

2.4.1 데이터 가용성에 관한 유의사항(A note on data availability)

섹션 2.2에서 언급한 바와 같이, 스팟 시장(spot market)의 증명자(provers)는 자신의 판단에 따라 요청에 참여하며, 데이터가 제공되지 않은 요청은 자유롭게 무시할 수 있다(각주 3, 5 참조).

하지만 서비스 계약(service agreements)의 경우 상황이 다소 복잡할 수 있다. 예를 들어, Alice가 Charlie로부터 서비스 계약을 구매했다고 가정해 보자. 이 계약에 따르면 Charlie는 Alice의 요청을 반드시 이행해야 하며, 이를 이행하지 못할 경우 Charlie는 페널티(slash)를 받게 된다. 그런데 Alice가 Charlie에게 요청을 제기한 후 실제로 해당 요청을 처리하는 데 필수적인 데이터를 제공하지 않았다면 어떤 상황이 발생할까? 일반적으로 Charlie가 Alice가 데이터를 제공하지 않았다는 사실을 증명할 방법이 없으므로, 서비스 계약의 스마트 컨트랙트 입장에서는 Alice가 유효한 요청을 했지만 Charlie가 이를 이행하지 못한 것으로 간주될 수 있다. Alice가 이를 악용하면, 실제로는 잘못이 없는 Charlie가 부당하게 페널티를 받게 되는 상황이 발생할 수 있다.

이러한 위험을 방지하기 위해, 증명자는 서비스 계약에 참여하기 전에 계약상 내포된 데이터 가용성(data availability) 요건을 면밀히 검토해야 한다. 예컨대 해당 계약이 이더리움(Ethereum), EigenDA, Celestia와 같은 허가가 필요 없는(permissionless) 공개적 소스에서 획득할 수 있는 데이터만으로 요청을 처리하도록 요구한다면, 증명자가 데이터의 가용성 문제로 인해 피해를 볼 가능성은 현저히 낮아진다. 다행스럽게도 대부분의 탈중앙화 애플리케이션들은 이러한 요구사항을 자연스럽게 충족하게 되는데, 이는 데이터의 가용성과 애플리케이션의 진행(progress) 사이의 관계가 비단 ZK에 국한된 문제만이 아니기 때문이다.

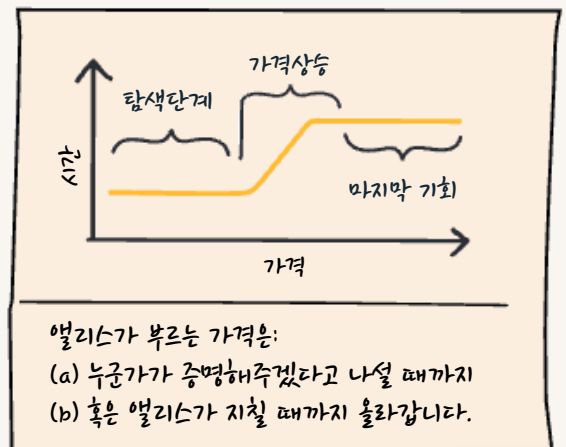
3 구현(Implementation)

바운드리스 프로토콜은 공개적으로 개발되고 있다. 요청자(requestor), 증명자(prover), 개발자(developer)를 위한 문서 및 코드 자체는 <https://beboundless.xyz>에서 확인할 수 있다.

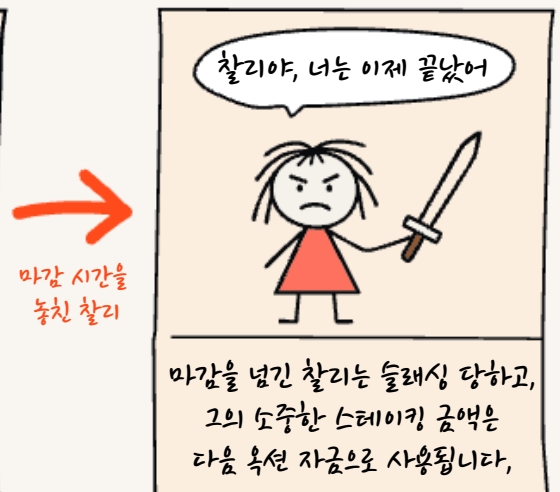
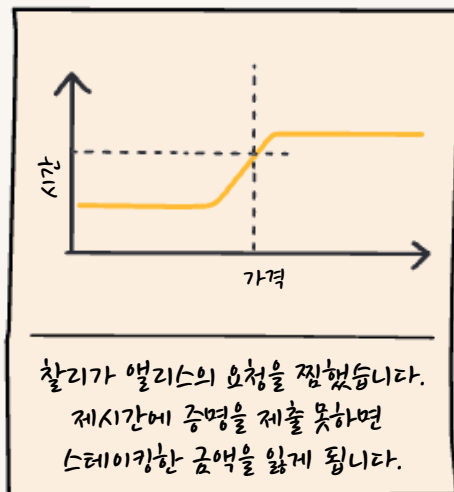
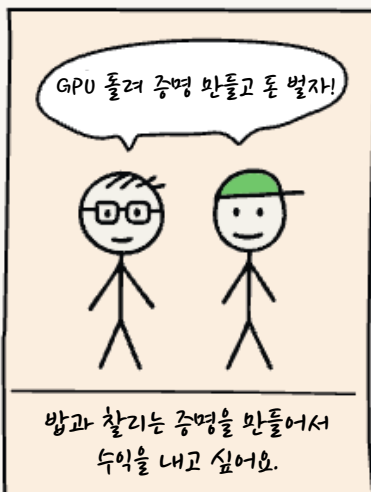
4 결론(Conclusion)

사토시가 그의 9페이지짜리 걸작 논문을 발표한 지 15년이 넘는 세월이 흘렀다. 그동안 많은 변화가 있었지만, 여전히 변하지 않은 것들도 있다. 오늘날 대부분의 온체인(on-chain) 생태계는 여전히 20세기의 암호학 기술을 근간으로 하고 있다. 최근 영지식 증명(zero knowledge proofs, ZK)의 발전은 이러한 생태계를 한 단계 더 도약시키는 중대한 진전을 이루었으며, 우리는 바운드리스 프로토콜이 이러한 혁신의 잠재력을 가장 이상적으로 실현하는 방법이라고 제안한다.

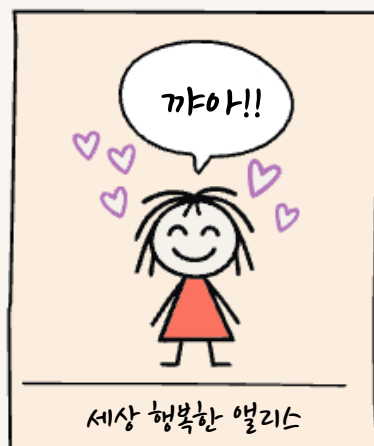
요청자



증명자



결과



제때 증명을 전달한 찰리

대체 옥션으로 무사히 해결!