

에이피아이쓰리 (Api3)

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.

초록

탈중앙화 금융(DeFi)과 같은 분야에서 탈중앙화 애플리케이션이 의미 있는 서비스를 제공하기 시작함에 따라, 이러한 애플리케이션이 기존 Web API를 통해 데이터를 수신하거나 이벤트를 트리거할 필요성이 점점 커지고 있습니다. 그러나 범용 오라클 솔루션은 과도하게 일반화되고 잘못된 접근 방식으로 인해 API 연결 문제를 적절히 해결하지 못하고 있습니다. 이 문제를 해결하기 위해 API3는 블록체인 네이티브 탈중앙화 API, 즉 dAPI라고 불리는 차세대 API를 구축하기 위한 협력적 노력을 주도하고 있습니다. dAPI는 API 제공자가 직접 운영하는 퍼스트 파티 오라클로 구성되며, 따라서 중개인을 활용하는 대안적 솔루션보다 더 안전하고 비용 효율적입니다. 이 이니셔티브의 거버넌스, 보안, 가치 포착 메커니즘의 핵심에는 API3 토큰이 있습니다. 토큰을 스테이킹하면 API3 DAO에 대한 거버넌스 권한과 모든 관련 보상을 획득할 수 있습니다. 스테이킹된 API3 토큰은 온체인 서비스 보증의 담보로 사용되며, 이를 통해 dAPI 사용자에게 정량화 가능하고 무신뢰성 보안을 제공합니다. 이러한 메커니즘은 생태계 차원에서 중앙 권한의 필요성을 제거할 것입니다. 결과적으로 API3 프로젝트는 스마트 컨트랙트 플랫폼이 API를 활용하여 진정으로 탈중앙화되고 신뢰 최소화 방식으로 의미 있는 애플리케이션을 구축할 수 있도록 합니다.

1. 서론

현실 세계와 상호작용할 수 있는 탈중앙화 애플리케이션의 등장하고 있으며, 이는 곧바로 가치에 반영되고 있습니다. 이 현상의 가장 대표적인 예는 2021년 11월 기준으로 다양한 애플리케이션에 \$100B 이상의 총 잠금 가치(TVL, Total Value Locked)가 유입된 탈중앙화 금융(DeFi, decentralized finance)의 급격한 성장입니다 [1]. 일반적인 DeFi 애플리케이션은 데이터 피드를 통해 자산 가격이 스마트 컨트랙트 플랫폼에 전달되기를 요구합니다 [2]. 이러한 데이터 피드는 애플리케이션이 현실 세계와 상호작용할 수 있도록 하며, 궁극적으로 파생상품 거래소나 대출과 같은 의미 있는 서비스를 제공할 수 있게 합니다. 현재 전개되고 있는 것은 단순히 DeFi의 부상만이 아니라, 현실 세계와 의미 있게 상호작용할 수 있는 탈중앙화 애플리케이션

선의 부상이며, DeFi는 빙산의 일각에 불과합니다.

기업들은 자산 가격 데이터를 제공하는 것부터 전통적 금융 거래를 실행하는 것에 이르기까지, 다양한 서비스를 Web API를 통해 제공하고 있습니다. 탈중앙화 애플리케이션이 현실 세계와 상호작용하기 위해서는 Web API가 제공하는 서비스에 접근할 수 있는 것이 필수적이지만, 이러한 API는 탈중앙화 애플리케이션과 네이티브하게 호환되지 않습니다. 기존의 중개인 기반 인터페이싱 솔루션은 중앙화되어 있고, 안전하지 않으며, 비용이 많이 듭니다. 또한 더 나은 대안이 없기 때문에 사용되고 있을 뿐입니다. API3는 API라는 개념이 제3자 중개인 없이도 Web 3.0의 필연적으로 엄격한 탈중앙화 요구사항을 충족하기 위한 진화적 다음 단계로 발전하도록 하는 것을 목표로 하고 있습니다. 이러한 차세대 탈중앙화 API는 dAPI라고 명명됩니다. dAPI는 전통적인 API 서비스를 스마트 컨트랙트에 탈중앙화된 방식으로 제공하기 위한 안전하고 비용 효율적인 솔루션입니다. dAPI는 다음과 같은 요소로 구성됩니다:

- 다수의 API (여기서 API라는 용어는 단순한 기술적 인터페이스뿐만 아니라 현실 세계의 비즈니스가 제공하는 서비스까지를 의미합니다.)
- 탈중앙화된 퍼스트 파티 오라클 네트워크. 즉, API 제공자 자체가 직접 운영하는 오라클입니다.
- 오라클 네트워크를 관리·감독하는 탈중앙화된 거버넌스 주체.

API3는 대규모로 dAPI를 구축·운영·수익화하기 위한 협력적 노력입니다. 이를 완전히 탈중앙화된 방식으로 달성하기 위해, 참여자들의 인센티브는 API3 토큰의 거버넌스, 보안, 가치 포착 기능을 통해 조정됩니다. 프로젝트는 완전히 개방적이고 직접적인 거버넌스 모델을 채택하고 있으며, 모든 API3 토큰 보유자는 스테이킹을 통해 API3 DAO에서 직접적인 의결권을 획득할 수 있습니다. 또한 스테이킹 참여자는 인플레이션 기반의 스테이킹 보상을 통해 dAPI 사용으로부터 이익을 얻을 수 있습니다.

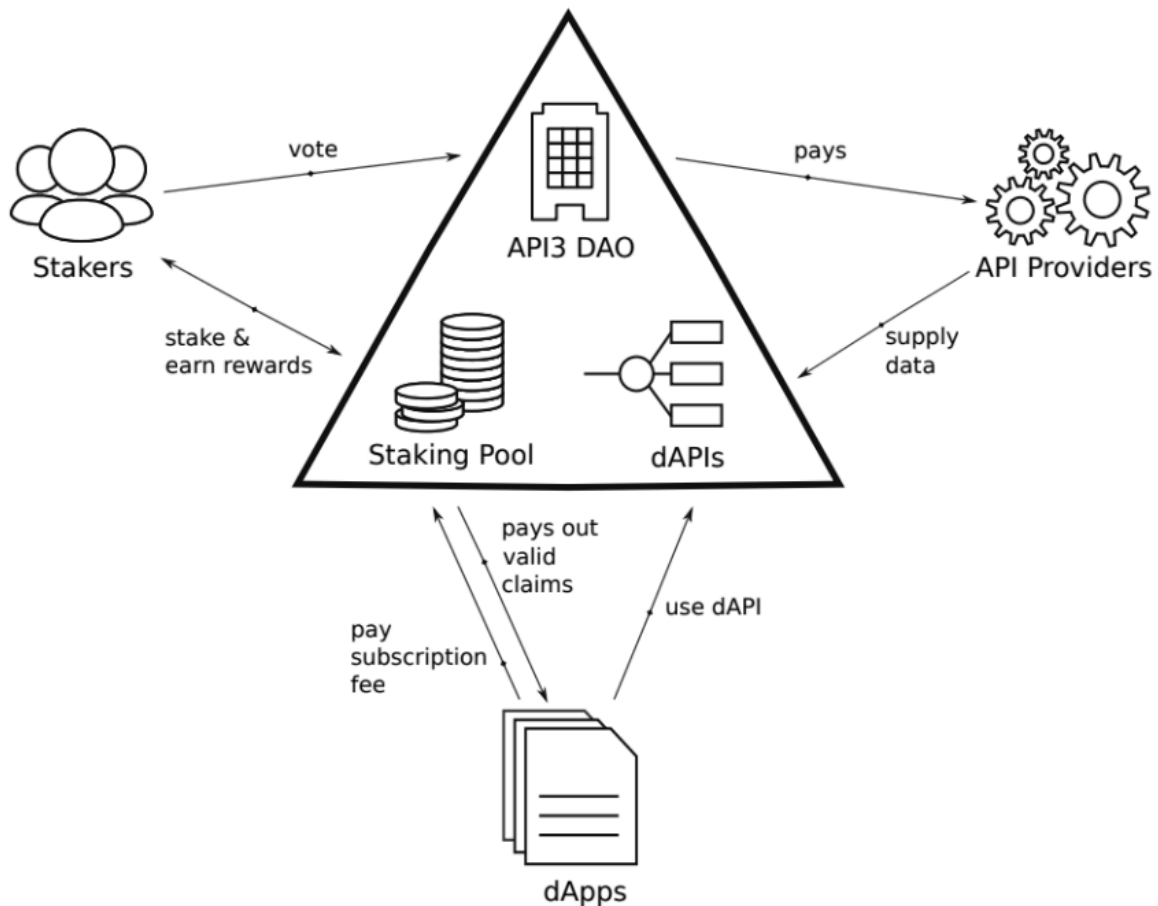


그림 1: API3 메커니즘 개요.

이와 더불어, 향후 DAO가 결정할 수 있는 추가적인 혜택 또한 제공될 수 있습니다. 스테이킹된 API3 토큰은 온체인 서비스 보증의 담보로 활용되어, dAPI 사용자에게 적격 오작동에 대해 정량화 가능하고 무신뢰성 보안을 제공합니다(그림 1 참조).

기존 오라클 솔루션의 근본적인 결함 중 하나는 데이터 소스와 기생적인 연결을 설정하고 유지하려는 시도에 있으며, 이는 지속 가능한 생태계를 만들어낼 수 없습니다. 이에 반해 API3는 API 제공자가 프로젝트의 동력이라는 인식에서 출발합니다. 따라서 API 제공자는 추상화되어 배제되는 것이 아니라 명시적으로 인정받고 보상받아, 그들의 이해관계가 API3 생태계 전체의 이해관계와 완전히 일치하도록 설계됩니다. 이미 API 제공자들이 유료 API의 무료 테스트넷 호출을 제공하거나 [3], 해커톤에 현금 보상을 제공하는 방식으로 [4] 탈중앙화 애플리케이션이 자사 서비스를 채택하도록 장려하는 모습을 확인한 바 있습니다. 이러한 협력을 더욱 심화시키는 것이 API3의 주요 강점 중 하나입니다.

탈중앙화 오라클 네트워크 솔루션은 API 제공자가 직접 오라클 노드를 운영하는 것이 종종 현실적으로 불가능하기 때문에 제3자 오라클을 활용합니다. 이는 제3자 오라클을 비용이 많이 드는 중개인으로 만들고, 추가적인 공격 벡터를 형성하게 됩니다. 이러한 문제를 제거하고 API 제공자가 생태계에 더욱 적극적으로 참여할 수 있도록 하기 위해, API3 데이터 피드는 API 제공자가 직접 운영하는 퍼스트 파티 오라클로 구성됩니다. 이는 Airnode를 통해 가능해지는데, Airnode는 블록체인 전문 지식, 유지관리, 운영을 API 제공자가 수행할 필요가 전혀 없는 완전한 서버리스(serverless) 오라클 노드입니다. 이로 인해 생성되는 dAPI는 비용 효율적이며 제3자 중간 계층으로부터의 공격에 대해 안전합니다.

오작동이 발생할 경우, dAPI 사용자는 사전에 합의된 한도 내에서 스테이킹 풀로부터 보상을 청구할 수 있습니다. 온체인 분쟁 해결 프로토콜인 Kleros [5]가 제시된 증거를 바탕으로 서비스 보증 약관에 따라 해당 청구가 지급되어야 하는지를 판정하는 데 사용됩니다. 이는 스테이킹 참여자가 dAPI가 투명하게 운영되고 보안 위험이 최소화되도록 거버넌스에 적극적으로 참여하도록 유도합니다. 성공적인 거버넌스—즉, 서비스 보증 청구로 인한 손실을 피하면서 dAPI를 성공적으로 수익화하는 것—은 API3 토큰으로 보상되며, 이는 거버넌스를 지속적으로 개선하는 긍정적 피드백 루프를 형성합니다.

그림 1을 다시 참조하면, API3 솔루션의 개요를 확인할 수 있습니다. dAPI는 전통적인 API 서비스를 탈중앙화되고 블록체인 네이티브한 방식으로 제공하는 퍼스트 파티 오라클 네트워크입니다. API3 DAO는 dAPI를 대규모로 구축·운영·수익화합니다. 탈중앙화 애플리케이션은 API3 토큰을 사용하여 dAPI에 접근합니다. API3 토큰 보유자는 스테이킹 풀에 참여하여 DAO에서 보상과 의결권을 획득합니다. 이 스테이킹 풀은 온체인 서비스 보증의 담보로 사용되어 dAPI 사용자에게 정량화 가능한 수준의 보안을 제공합니다. API3는 탈중앙화, 비용 효율성, 보안, 투명성, 그리고 생태계 성장 잠재력 측면에서 기존 오라클 솔루션을 개선합니다.

2. API 연결 문제

애플리케이션 프로그래밍 인터페이스(API)는 특정 애플리케이션과 통신하여 그로부터 서비스를 제공받기 위해 사용되는 잘 표준화되고 문서화된 프로토콜입니다. 이러한 서비스는 데이터를 수신하는 형태일 수도 있고 이벤트를 트리거하는 형태일 수도 있습니다. 애플리케이션은 API를 통해 서로 통신할 수 있으며, 이를 통해 서로 통합되어 더 복잡한 애플리케이션을 구축할 수 있습니다. 이러한 이유로 API는 디지털 세계를 이어 주는 접착제라고 불립니다.

기업들이 API를 활용하여 데이터와 서비스를 수익화함에 따라, API라는 개념은 초기 의미를 넘어 확장되었습니다. 이제 API라는 용어는 단순히 인터페이스의 기술적 구현을 지칭하는 것이 아니라, 그 안에 포함된 서비스를 포괄하는 완전한 제품을 의미합니다 [6]. API를 제공하는 글로벌 기업들은 평균적으로 조직 수익의 25%를 API에서 창출하고 있으며 [7], Salesforce, Expedia, eBay와 같은 기업들은 대부분의 수익을 API를 통해 얻는다고 보고했습니다 [8]. 이는 전적으로 API 중심적인 비즈니스 모델로의 전환이 임박했음을 보여줍니다 [9]. 심지어 은행업과 같은 전통적인 산업조차도 이 변화에 의해 재편될 것으로 예상됩니다 [10].

기존 서비스를 API를 통해 애플리케이션에 통합할 수 있게 되면서, 개발자들은 점점 더 복잡하고 강력한 애플리케이션을 구축할 수 있었고, 이는 거대한 웹 서비스와 모바일 애플리케이션의 성장을 이끌었습니다. 그러나 기업들이 서비스를 제공하기 위해 사용하는 이러한 API는 2.1절에서 설명될 기술적 이유로 인해 스마트 컨트랙트와 직접적으로 호환되지 않으며, 이는 의미 있는 탈중앙화 애플리케이션의 발전을 저해해 왔습니다. 따라서 현실 세계와 상호작용할 수 있는 탈중앙화 애플리케이션을 구축하는 데 직면한 난제는 API 연결 문제(API connectivity problem)라고 가장 잘 표현될 수 있습니다. 이 문제를 잘못 해석하는 경우, 최적 이하의 해결책으로 이어질 수 있습니다.

2.1 오라클 문제: 데이터 출처에 구애받지 않는(source-agnostic) 오해

탈중앙화는 Web 3.0을 정의하는 핵심 개념으로, 이는 계산을 분산하고 사전에 정해진 합의 규칙을 통해 결과를 확정짓는 방식으로 설명됩니다 [11]. 탈중앙화 애플리케이션의 비즈니스 로직은 스마트 컨트랙트로 구현되며 [12], 이는 블록체인 기반 스마트 컨트랙트 플랫폼에서 실행됩니다 [13]. 탈중앙화는 참여자들이 상호 신뢰나 신뢰할 수 있는 제3자를 필요로 하지 않고 협력할 수 있도록 하여, 공격과 검열에 대한 강인성을 제공합니다.

합의 규칙을 강제하기 위해, 스마트 컨트랙트 플랫폼의 노드들은 각 컨트랙트 호출이 올바른 결과를 산출했는지 확인하기 위해 해당 계산을 로컬에서 반복 실행해야 합니다. 이를 가능하게 하기 위해, 스마트 컨트랙트는 모든 노드가 접근할 수 있고 합의할 수 있는 정보만을 활용할 수 있습니다. 간단히 말해, 스마트 컨트랙트는 블록체인에 이미 존재하는 정보만 다룰 수 있으며, 외부 세계와 직접 상호작용할 수 없습니다. 이는 블록체인에 임의로 정의된 진실을 전달할 수 있는 이상화된 주체를 지칭하는 “오라클 문제(oracle problem)”로 널리 알려져 있습니다.

오라클 문제는 그 이름 자체가 불가능한 해결책을 시사하듯, 잘못 제기된 문제입니다. 비유하자면 A 지점에서 B 지점으로 이동하는 문제를 “순간이동 문제”로 접근하는 것과 같습니다. 그럼에도 불구하고 1세대 솔루션들은 문자 그대로의 오라클을 구현하려 시도했는데, 이는 질문을 제기하고 그 답변을 클라우드소싱하는 방식이었습니다. 이러한 방식은 며칠 단위의 응답 시간과 막대한 트랜잭션 수로 인해 발생하는 극단적인 가스 비용을 초래하며 [14], 이는 DeFi나 예측 시장과 같은 사용 사례에 적합하지 않습니다. 다만 이 접근 방식은 제공해야 하는 정보가 주관적인 경우에는 적절합니다. 예로는 사법 분쟁의 해결을 들 수 있습니다 [5].

2세대 솔루션은 범위를 축소하여, 프로그래밍적으로 접근할 수 있는 사실적 정보만을 다루도록 했고, 그 결과 “상호운용성 문제(interoperability problem)”에 도달하게 되었습니다. 이들 솔루션에서는 블록체인과 API, 또는 두 개의 블록체인처럼 임의의 두 시스템에 통합된 오라클 노드가 양자 간의 자동화된 중개자로 작동합니다 [15–18]. 여러 오라클 노드를 활용하고 사전에 정해진 합의 규칙을 통해 결과를 확정함으로써, 기초가 되는 블록체인 기술을 보완하는 보안적 보장을 제공합니다. 이러한 솔루션은 클라우드소싱 오라클에 비해 더 빠르고 비용도 적게 들어 더 많은 사용 사례에 적합하지만, 근본 문제에 대한 과도한 일반화에 기반해 설계되었다는 한계를 지닙니다.

상호운용성 솔루션은 API 제공자, 오라클, 데이터 소비자라는 세 주체를 포함합니다 [19]. 그러나 이들은 생태계를 오직 오라클과 데이터 소비자만으로 구성된 것으로 모델링하는 오류에 빠지며, 데이터의 기원이 어디인지 간과합니다. 다시 말해, 이들 모델은 오라클 노드를 진리의 원천인 신화적 오라클처럼 취급합니다. 문제의 3분의 1을 무시하는 이러한 접근은 실행 불가능한 해결책을 실행 가능한 것처럼 보이게 만듭니다.

상호운용성 솔루션이 데이터 출처에 구애받지 않는(source-agnostic) 방식으로 설계된 결과는 다음과 같습니다:

- API 제공자가 직접 운영하는 오라클로 대체될 수 있었음에도 불구하고, 안전하지 않고 비용이 많이 드는 제3자 오라클의 중간 계층이 존재합니다.
- 실제 데이터 출처는 배제된 채, 임대 수익을 추구하는 중개인(rent-seeking middlemen)을 양성하는 생태계가 형성됩니다.

- 데이터 피드 내에서 서로 다른 출처로부터 수신된 데이터가 무차별적으로 동일하게 취급됩니다.

상호운용성 솔루션에서 만연한 또 다른 문제는, 이들이 로우레벨(low-level) 프로토콜이기 때문에 인터페이스를 완전한 제품이 아닌 기술적 구성 요소, 즉 *미들웨어(middleware)* 로 간주한다는 점입니다. 그 부작용으로 인터페이스의 거버넌스가 범위에서 제외됩니다. 그러나 거버넌스는 결코 사소한 문제가 아닙니다. 탈중앙화된 인터페이스는 독립적이고 경쟁적인 주체들이 협력해야 하기 때문입니다. 현재 활용되는 해결책은 신뢰할 수 있는 중앙화된 기관이 인터페이스를 관리하는 것이며, 이는 탈중앙화라는 근본 목표와 정면으로 배치됩니다. 관리 기관이 오라클 네트워크의 출력을 완전히 통제하게 되므로, 중앙화된 거버넌스를 가진 탈중앙화 오라클 네트워크는 사실상 몇 가지 절차만 추가된 중앙화 오라클에 불과합니다.

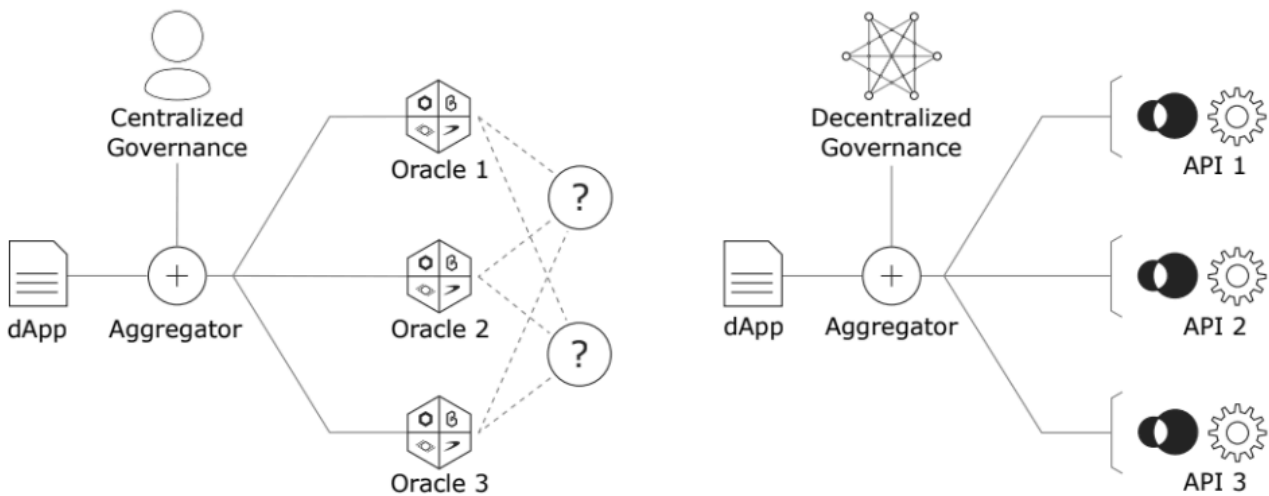


그림 2: 탈중앙화 상호운용성 솔루션은 출처를 네이티브하게 공개하지 않는 제3자 오라클을 사용합니다. 반면 dAPI는 API 제공자가 직접 운영하는 Airnode로 구성된 퍼스트 파티 오라클로 이루어져 있습니다. 또한 dAPI는 거버넌스 방식에서도 탈중앙화를 구현하여, 엔드-투-엔드(end-to-end) 탈중앙화를 달성합니다.

2.2 탈중앙화 API

이전 세대 상호운용성 솔루션의 문제는 새로운 관점을 도입해야만 해결될 수 있습니다. 본질적으로 현재 문제는 탈중앙화 애플리케이션이 기존 API 제공자로부터 서비스를 탈중앙화된 방식으로 제공받을 수 없다는 데 있습니다. 실제로 오늘날 상호운용성 솔루션의 주요 활용 사례는 중앙화 거래소 API로부터 자산 가격을 DeFi 애플리케이션에 전달하는 것이며, 예측 시장 [20]이나 파라메트릭 보험 [21]과 같은 신흥 활용 사례 역시 유사한 요구사항을 가지고 있습니다. 따라서 문제 정의를 이와 같이 구체화하면 차세대 현실 세계 상호연결성 솔루션에 도달할 수 있습니다.

이 새로운 문제 정의는 탈중앙화 애플리케이션이 특정 Web API 서비스를 블록체인에 전달받을 필요가 있으며, 이는 완전한 탈중앙화 방식으로, 비용 효율적이고 안전하게 이루어져야 함을 의미합니다. 이러한 요구사항을 규정함으로써 이를 최적의 방식으로 충족하는 완전한 제품을 설계할 수 있습니다. 즉, 탈중앙화 API(dAPI)는 API 제공자가 직접 운영하는 퍼스트 파티 오라클 네트워크이며, 탈중앙화된 방식으로 거버넌스가 이루어집니다. 반대로, 탈중앙화 상호운용성 솔루션은 제3자 중개인으로 이루어진 오라클 네트워크와 중앙화된 기관이 운영하는 거버넌스 구조로 구성되며, 이는 불완전한 문제 정의로 인해 필연적으로 나타나

는 결과입니다. (그림 2 참조)

3. 제3자 오라클을 중개인으로 두는 문제점

기존 솔루션들은 임의의 시스템이 또 다른 임의의 시스템과 기술적 인터페이스를 통해 상호운용 가능해야 한다는 단순화된 문제를 전제로 합니다. 이러한 과도한 일반화는 제3자 오라클만이 지원할 수 있는 유연한 인터페이스를 요구합니다. 그러나 실제 문제의 범위는 훨씬 제한적이기 때문에 이는 최적의 해결책이 아닙니다. 대부분의 경우, 탈중앙화 상호운용성 문제는 기존 API 제공자로부터 서비스를 탈중앙화된 방식으로 제공받는 문제일 뿐입니다. 이러한 보다 제한된 문제 정의는 인터페이스 경로에 제3자 중간 계층을 필요로 하지 않는 최적의 해결책을 가능하게 합니다. 이하에서는 상호운용성 솔루션에서 중개인에 의존하는 것이 초래하는 결과를 논의합니다.

3.1 취약성(Vulnerability)

탈중앙화 오라클 네트워크는 오라클 보고서를 단일 답으로 축소하기 위해 집계 함수(aggregation function)를 사용합니다. 이 함수는 본질적으로 합의 알고리즘이며, 모든 합의 알고리즘과 마찬가지로 일정 비율의 불성실한 참여자에 취약합니다. 즉, 악의적인 오라클 그룹이 결탁하여 결과를 왜곡하거나 완전히 통제할 수 있습니다. 또한 단일 행위자가 다수의 오라클 노드 운영자 신원을 위조하고, 일정 기간 동안 성실한 운영 실적을 쌓아 동일한 유형의 공격을 독자적으로 수행할 수도 있는데, 이는 시빌 공격(Sybil attack)이라 불립니다 [22].

인터페이스 경로에 추가적인 참여자 계층을 두는 가장 치명적인 단점은 새로운 공격 벡터가 형성된다는 점입니다. 이는 각 중개인 계층이 앞서 언급한 결탁 공격이나 시빌 공격을 독립적으로 실행할 수 있음을 의미합니다. 따라서 보안 측면에서 궁극적인 해결책은 중개인을 완전히 제거하는 것입니다.

3.2 중개인 세금(Middleman Tax)

오라클은 정직하게 보고하거나 허위 보고(서비스 거부 포함)를 할 수 있는 게임을 수행합니다. 정직하게 보고하는 경우 오라클이 얻는 보상은 점진적이지만, 게임을 계속 이어갈 수 있습니다. 반면 허위 보고의 경우 보고 결과에 의존하는 계약이 확보하는 가치에 비례한 일회성 보상을 얻을 수 있지만, 게임은 종료됩니다(그림 3 참조). 따라서 트랜잭션 t_i 시점부터 오라클이 얻을 수 있는 최대 누적 보상은 다음과 같이 정의됩니다:

$$P[i] = \max(x_i, v_i + P[i + 1]). \tag{1}$$

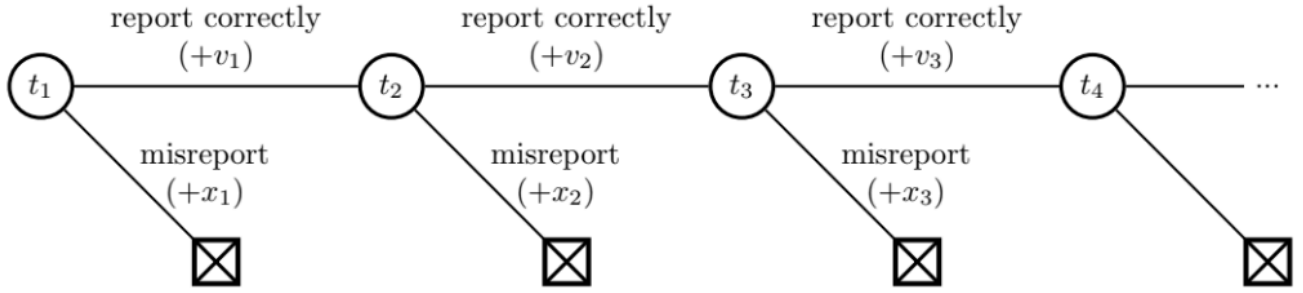


그림 3: 오라클이 취할 수 있는 행동과 그 결과를 설명하는 의사결정 트리. 특정 트랜잭션 t_i 에서 오라클은 정직하게 보고하여 v_i 를 얻거나, 허위 보고하여 x_i 를 얻을 수 있습니다. 불성실한 행동은 오라클이 더 이상 사용되지 않음을 의미하며, 즉 게임이 종료됩니다.

합리적인 오라클은 공격을 통해 얻을 수 있는 이익이 공격을 수행하지 않았을 경우 얻을 수 있는 잠재적 이익보다 클 경우 결국 허위 보고를 하게 됩니다. 즉, 합리적인 오라클에 대해 다음 조건이 성립한다면, 오라클은 언젠가 허위 보고를 하게 됩니다:

$$\exists i \in \mathbb{N}, ; x_i > v_i + P[i + 1]. \quad (2)$$

이는 오라클이 정직하게 행동하여 얻을 수 있는 잠재적 이익이 항상 허위 보고로부터 얻을 수 있는 이익보다 커야만 허위 보고를 피할 수 있음을 의미합니다. v 를 오라클이 요청당 지급받는 보상으로, x 를 오라클의 응답으로 확보되는 금액으로 근사할 수 있으나, 이는 위험을 과소평가하는 것입니다. 왜냐하면 오라클이 허위 보고를 하도록 기울게 만드는 추가적인 요인들이 존재하기 때문이며, 그 예시는 다음과 같습니다:

- 시간 선호 이론(time preference theory)에 따르면 [23], 오라클 노드 운영자는 미래 보상을 덜 가치 있게 평가합니다(즉, i 가 증가할수록 v_i 는 감소합니다).
- 실제로 오라클이 정직하게 행동한다고 해서 게임이 반드시 지속된다는 보장이 없으며, 이 위험은 미래 보상의 가치를 더욱 감소시킵니다.
- 공격을 수행함으로써 추가적인 이익을 얻을 수도 있습니다. 예를 들어, 오라클 솔루션의 실패로 자산이 평가절하될 때 해당 자산에 숏 포지션을 취하는 경우입니다.

이러한 불확실성으로 인해 요구되는 v_i 는 과대평가될 필요가 있으며, 즉 오라클이 공격을 하지 않도록 하기 위해서는 과도하게 높은 비용을 지불해야 합니다.

이 모델은 탈중앙화 오라클 네트워크로 확장될 수 있습니다. 오라클 보고서 또는 그 산출물이 온체인에 기록되기 때문에, 결탁한 오라클에게 무신뢰성 보상을 지급하는 스마트 계약을 구현하는 것은 매우 간단합니다. 이는 곧 공격으로부터 이익을 얻을 수 있는 제3자가 오라클을 고용하여, 그들이 결탁할 경우 결정적인 보상 금액을 확실히 지급받을 수 있다는 의미입니다.

거시적으로 볼 때, 오라클의 역할은 본질적으로 (1) 온체인 요청을 수신하고, (2) 해당 오프체인 API 호출을 수행하며, (3) 그 응답을 다시 온체인에 기록하는 것입니다. 따라서 제3자 오라클은 근본적으로 중개인입니다. 제공되는 서비스가 가능한 한 최소한의 것이지만, 앞서 설명한 이유로 인해 이러한 중개인은 데이터 피드가 보호하는 가치에 비례한 대가를 지급받아야 합니다. 이는 특히 DeFi와 같은 고가치 사용 사례에서 심각한

문제가 됩니다. 이 현상을 “중개인 세금(middleman tax)”이라고 부르며, 제3자 오라클을 배제함으로써 이를 완전히 제거할 수 있고, 사용자에게 매우 큰 비용 절감을 가져옵니다.

3.3 비효율적인 중복성

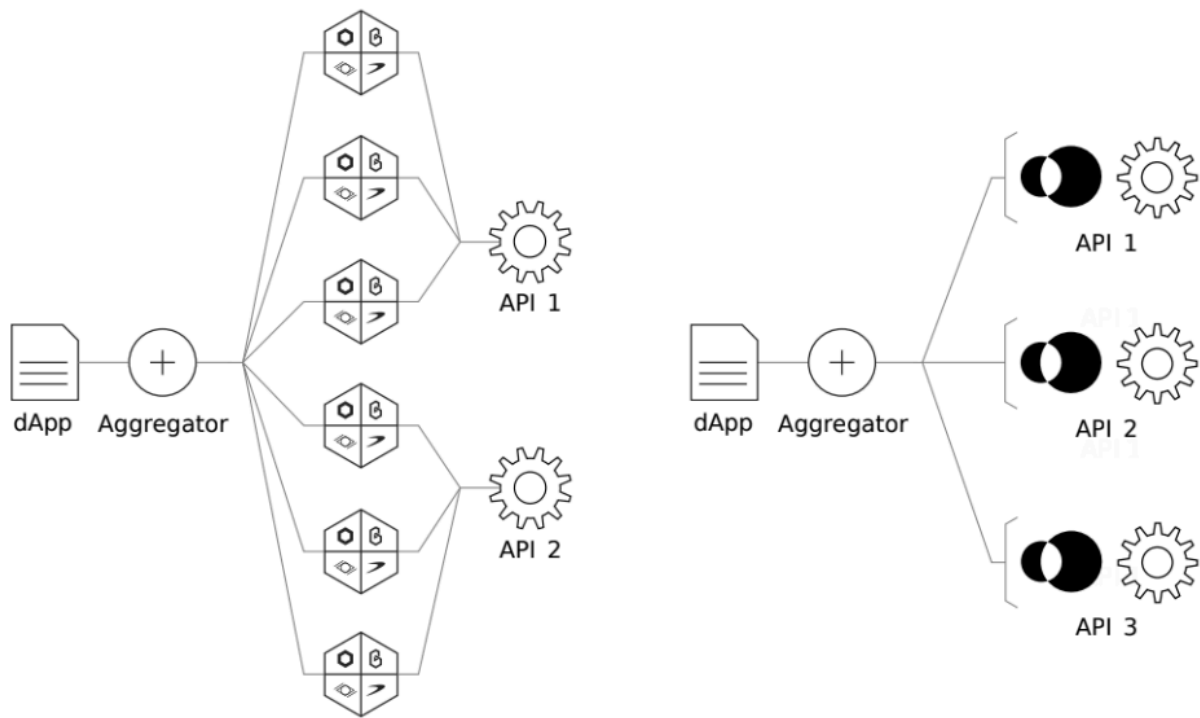


그림 4: 제3자 오라클을 사용하는 경우 오라클 수준에서 불필요하게 중복된 탈중앙화를 요구하게 됩니다. 반면 퍼스트 파티 오라클은 더 안전하고 비용 효율적인 방식으로 더 나은 수준의 탈중앙화를 제공합니다.

제3자 오라클에 의존하는 데이터 피드는 오라클 수준에서 과도한 중복성을 요구합니다(그림 4 참조). 이는 제3자 오라클이 API 제공자보다 신뢰성이 훨씬 떨어지기 때문이며, API 제공자는 전통적인 오프체인 비즈니스와 그에 따른 평판을 유지해야 하기 때문입니다. 일반적으로 하나의 데이터 피드에서 각 API 제공자는 2–3개의 오라클에 의해 서비스됩니다. 그러나 이러한 탈중앙화는 데이터 소스 수준에서의 추가적인 보안을 제공하지 않으며, 단지 제3자 오라클 사용으로 인한 취약성을 완화할 뿐입니다. 불행히도 이는 운영 비용을 여러 단계에서 증가시키는 결과를 낳습니다. 예를 들어, 데이터 피드는 본질적으로 오라클 노드를 운영하는 모든 기술 인력을 고용하는 것과 같으며, 노드 수가 많아질수록 더 많은 인원을 지원해야 합니다. 또한 더 많은 오라클을 사용하는 것은 가스 비용의 직접적인 증가로 이어집니다. 구체적으로, 오라클 요청–응답의 가스 비용은 오라클 수에 선형적으로 증가하며, 중앙값(median)과 같이 조합 연산을 수행하는 집계 함수의 가스 비용은 초선형적으로 증가합니다.

3.4 투명성 부족

API 수준에서의 탈중앙화와 오라클 수준에서의 탈중앙화는 서로 독립적이며, 전체 시스템은 둘 중 더 중앙화된 부분, 즉 가장 약한 연결부위(link)에 의해 좌우됩니다. 그러나 일반 대중은 물론 탈중앙화 오라클 네트워크의 사용자조차 이 사실을 간과하고, 오라클 수준의 탈중앙화를 시스템 전체의 탈중앙화와 혼동합니다. 이러한 오해는 주로 오라클이 사용하는 데이터 소스에 대한 투명성 부족 [24]에서 기인하며, 그 결과 탈중앙화가 데이터 소스(API) 수준에서 심각하게 병목되고 있다는 사실이 가려집니다.

제3자 오라클로 구성된 데이터 피드는 실제보다 더 탈중앙화된 것처럼 보입니다. 또한 데이터 피드가 데이터 출처 측면에서 투명하지 않을 경우, 개발자는 데이터 피드의 무결성을 평가할 수 없으며 거버넌스 주체를 신뢰할 수밖에 없습니다. 그러나 데이터 소스가 투명하지 않다면, 거버넌스 주체가 품질을 낮은 가격이나 편의성보다 우선시할 즉각적인 인센티브는 존재하지 않으며, 이는 흔히 “garbage in, garbage out”으로 불리는 결과를 초래할 수 있습니다.

흥미롭게도, 거버넌스 주체에게 유리한 전술인 데이터 소스의 불투명화는 제3자 오라클에게는 필수적이기까지 합니다. 대부분의 API 서비스 약관은 API 데이터의 재판매나 무단 배포를 금지하고 있으며, 이로 인해 해당 API를 제공하는 오라클 노드 운영자는 약관을 위반하게 되고 API 제공자의 청구를 포함한 광범위한 법적 책임에 노출될 수 있습니다 [25]. 이 문제는 API 호출 시간, 응답, 지불이 모두 퍼블릭 블록체인에 기록됨으로써 더욱 악화됩니다. 이는 개별 노드 운영자를 소송 위험에 노출시킬 뿐만 아니라, 대규모로 조정된 법적 조치가 가해질 경우 기존 제3자 오라클 네트워크 전체가 즉시 운영 불능 상태에 빠지며 신규 참여자 유입까지 차단하는 시스템적 위험을 초래합니다.

그러나 데이터 소스의 불투명성과 단순화는 관행일 뿐, 반드시 필요한 것은 아닙니다. 특히 API 제공자와 생태계의 인센티브가 일치할 경우, 오라클은 API 제공자의 명시적 동의를 통해 API 데이터를 사용자에게 제공하고, 데이터 소스를 사용자에게 공개할 수 있습니다 [3]. 이는 온체인 데이터 수요를 증가시키므로 API 제공자에게도 이익이 됩니다.

4. Airnode: 퍼스트 파티 오라클을 위해 설계된 노드

퍼스트 파티 오라클은 API3 솔루션의 핵심 구성 요소입니다. 이는 각 API가 제3자가 아닌 해당 API의 소유자인 주체가 직접 운영하는 오라클을 통해 제공된다는 것을 의미합니다. 본 절에서는 퍼스트 파티 오라클을 활용하는 이점과, API3가 Airnode를 통해 API 제공자가 직접 오라클을 운영할 수 있도록 하는 방법을 논의합니다.

4.1 중개인 제거의 이점

앞서 3장에서 논의한 모든 문제의 단순한 해결책은 바로 퍼스트 파티 오라클, 즉 API 제공자가 직접 운영하는 오라클입니다. API 제공자가 오라클을 직접 운영한다는 것은, 스마트 컨트랙트 플랫폼 프로토콜 수준에서 자신들의 프라이빗 키로 응답에 서명한다는 것을 의미하며, 이는 데이터가 변조되지 않았음을 보장하는 가장 확실한 증거입니다. 또한 제3자가 처리 중인 원시 API 데이터를 관찰할 수 없기 때문에, 퍼스트 파티 오라클

은 기본적으로 프라이빗하며, 이로 인해 더 다양한 활용 사례에서 네이티브하게 사용될 수 있습니다.

퍼스트 파티 오라클로 구성된 데이터 피드는 중개인을 고용한 경우보다 비용 측면에서 효율적입니다. 중개인에게 서비스 제공에 대한 비용뿐 아니라 데이터 피드 공격을 억제하기 위한 인센티브(3.2절에서 설명한 ‘중개인 세금’)도 지불해야 하기 때문입니다. 또한 퍼스트 파티 오라클 기반 데이터 피드는 제3자 공격으로부터 보호하기 위해 오라클 수준에서 과도한 탈중앙화를 필요로 하지 않으므로, 더 적은 수의 오라클만으로도 충분합니다. 일반적으로 각 API가 최소 두 개 이상의 제3자 오라클에 의해 제공된다는 점을 고려할 때, 퍼스트 파티 오라클이 지원하는 데이터 피드는 보수적으로 추정해도 가스 비용 측면에서 최소 50% 이상 효율적입니다.

퍼스트 파티 오라클은 데이터 소스와 탈중앙화 정도 측면에서 필요한 투명성을 제공합니다. 각 API 제공자가 오라클을 직접 운영하게 되며, 이 오라클은 온체인에서 확인 가능하므로, 특정 데이터 피드를 제공하는 오라클의 수는 탈중앙화 수준을 정확히 반영하게 됩니다. 이는 오라클과 데이터 소스 간의 일대일 매핑이 이루어지기 때문입니다. 또한 API 제공자는 오프체인 채널을 통해 온체인 신원을 공개하므로, 사용자는 자신이 소비하는 데이터의 출처를 언제든지 검증할 수 있습니다.

마지막으로 API 제공자가 오라클을 직접 운영하게 되면 3.4절에서 언급된 법적 문제 역시 해결됩니다. API 서비스가 제3자에게 라이선스될 필요가 없어지고, API 제공자가 전체 수익을 가져가게 되기 때문입니다. 더 나아가 이는 임대 수익만을 추구하는 제3자 오라클 문제를 해소하며, 자금을 실제로 가치 창출을 담당하는 주체인 API 제공자에게 재배분할 수 있습니다. API 제공자에게 인센티브를 제공함으로써 API3 생태계와의 경제적 이해관계가 일치하게 되며, 그 결과 양측 간의 강력한 상호 유대가 형성됩니다.

4.1.1 데이터의 오프체인 서명(Off-Chain Signing of Data)

여전히 제3자 오라클에 의존하면서도, 그들이 데이터를 변조하지 못하도록 하는 하이브리드 방식이 존재합니다. 이 방식에서 API 제공자는 오프체인에서 자신의 프라이빗 키로 데이터를 서명한 뒤, 이를 일반 API 엔드포인트를 통해 제공합니다. 제3자 오라클은 해당 엔드포인트를 호출하여 서명된 데이터를 가져와 온체인에 게시합니다. 이후 API 제공자의 퍼블릭 키를 활용하여 데이터가 제3자 오라클에 의해 변조되지 않았음을 온체인에서 검증할 수 있습니다 [26].

이 방식은 오라클 수준에서의 데이터 변조 위험을 제거하지만, 본질적으로는 불완전한 절충안에 불과합니다. 제3자 오라클에 의존하기 때문에 여전히 그로 인한 생태계 문제를 겪게 되며, 추가적으로 오프체인 서명을 구현하기 위해 API 측에서의 수정이 필요합니다. 이로 인해 활용 가능한 API 선택지가 기존 제3자 오라클 기반 솔루션보다도 훨씬 제한되며, 솔루션의 생태계 성장 잠재력은 애플리케이션 규모에 국한됩니다.

4.2 API 제공자가 오라클을 운영하는 데 따른 장벽

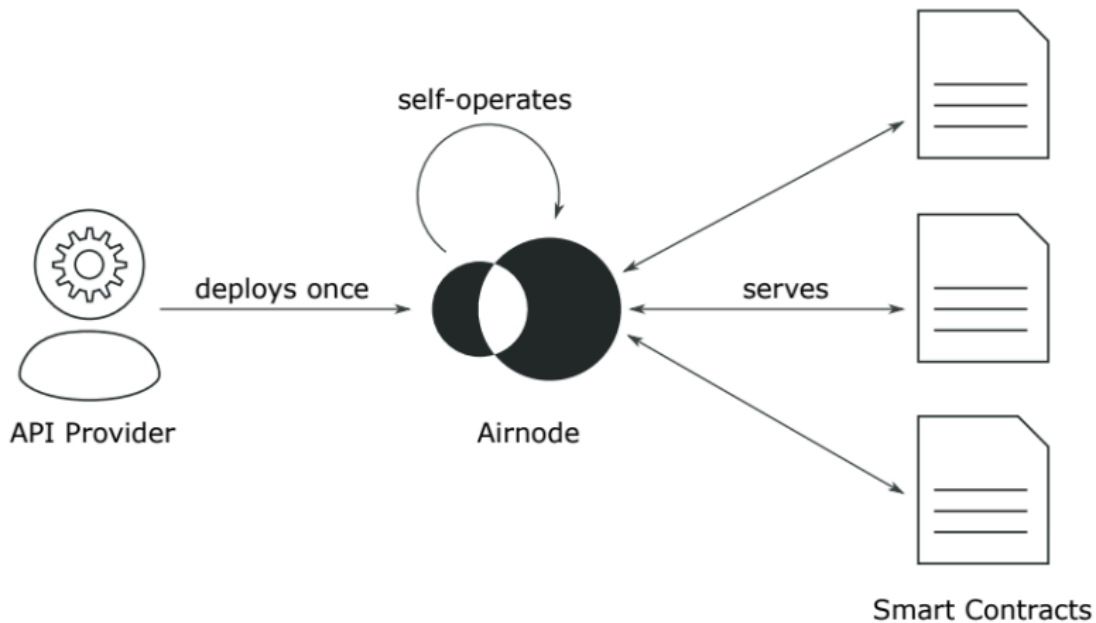


그림 5: Airnode는 API 제공자가 한 번 배포하면 이후 추가적인 유지보수를 요구하지 않도록 설계되었습니다.

Honeycomb API Marketplace [19] 관련 작업을 수행하며 API 제공자와 광범위하게 소통한 결과, 오라클 온보딩 및 운영에 다음과 같은 장벽이 존재함을 확인하였습니다:

1. 전통적인 API 제공자는 일반 대중보다 블록체인 기술에 더 익숙하지 않은 경우가 대부분입니다. 이는 암호화폐 시장 데이터를 큐레이션하는 제공자에게도 적용되는데, 그들의 주요 업무는 거래소 API에서 데이터를 수집하는 것이며, 이는 블록체인 관련 전문 지식을 필요로 하지 않기 때문입니다. 따라서 대부분의 API 제공자는 자체 자원만으로 오라클 노드를 즉시 운영할 수 없습니다.
2. 오라클 노드 운영자를 위한 인력 시장이 존재하지 않습니다. 일부 API 제공자가 소수의 노드 운영자를 채용하여 필요한 전문 지식을 확보할 수 있다고 해도, 이는 확장 가능한 해결책이 되지 못합니다.
3. 오라클 노드를 운영하는 데에는 인력 시간과 인프라 비용의 형태로 많은 자원이 소모됩니다. 상당한 보조금이나 미래 수익이 보장되지 않는 한, 오라클 노드 운영은 재정적으로 실행 불가능합니다.
4. 오라클 노드를 운영하려면 API 제공자가 암호화폐로 거래해야 합니다. 구체적으로는 네이티브 토큰(예: ETH)으로 가스 비용을 지불하고, 하나 이상의 암호화폐로 대금을 수령해야 합니다. 이로 인해 규제, 법적, 회계상의 이유로 대부분의 API 제공자는 참여할 수 없게 됩니다. 또한 API 제공자가 자금을 스테이킹하도록 요구하는 어떠한 방식도 유사한 재정적 위험 요인으로 인해 전면적으로 거부됩니다.

4.3 Airnode 특징

Airnode는 API 제공자가 자체 오라클을 직접 운영할 수 있도록 특별히 설계된 완전 서버리스(serverless) 오라클 노드입니다(그림 5 참조). 이는 4.2절에서 언급된 오라클 노드 관련 모든 문제를 해결합니다:

1. 운영을 위해 특정한 전문 지식을 요구하지 않습니다. 실제로 Airnode는 완전히 *설치 후 방치(set and forget)* 방식으로 설계되었기 때문에 “운영”이라는 개념조차 언급하기 어렵습니다.
2. 운영체제 업데이트나 가동 시간 모니터링과 같은 일상적 유지보수가 필요하지 않습니다. 이는 완전 관리형 서버리스 기술을 활용하기 때문입니다. 또한 상태 비저장(stateless) 방식으로 설계되어, 영구적 다운타임을 유발하고 운영자 개입을 요구할 수 있는 문제에 대해 매우 강인합니다.
3. 사용량 기반(on-demand) 서비스 위에 구축되어 있어, 노드 운영자는 노드가 사용된 만큼만 비용을 지불합니다. 이로 인해 API 제공자는 무료로 오라클을 실행할 수 있으며, 실제로 수익을 창출한 이후에만 비용을 지불하게 됩니다.
4. 노드 운영자가 암호화폐를 직접 다룰 필요가 전혀 없습니다. 프로토콜은 요청자가 모든 가스 비용을 부담하도록 설계되어 있습니다.

Airnode는 웹 API를 스마트 컨트랙트 플랫폼과 추가 비용이나 결제 토큰의 불편함 없이 연결할 수 있도록 하는 경량 래퍼(wrapper)로 이해할 수 있습니다. API 제공자 입장에서의 관여 수준을 고려하면, Airnode 사용은 블록체인 노드를 부업으로 운영하는 것이라기보다, API를 웹에서 접근 가능하게 만드는 API 게이트웨이를 활용하는 것과 유사합니다. 실제로 API3의 목표는 Airnode가 API 게이트웨이처럼 보편적이고 평범하게 자리잡는 것이며, 이를 통해 다양한 퍼스트 파티 오라클이 API3에서 활용 가능해지도록 하는 것입니다.

API 제공자는 고가용성(high availability) 인프라를 구축하기 위해 상당한 자원을 투자합니다. 따라서 오라클 노드 구현은 단일 장애 지점(single point of failure)을 포함하지 않아야 하며, 이는 다운타임을 초래할 수 있습니다. 기존 제3자 오라클을 활용하는 솔루션은 이를 보완하기 위해 오라클 수준에서 과도한 중복성을 의존했는데, 이는 3.3절에서 언급한 바와 같이 과도한 비용을 발생시킵니다. API3는 각 API가 해당 퍼스트 파티 오라클에 의해서만 서비스되도록 구상하며, 이에 따라 중복성은 개별 Airnode 수준에서 구현되어야 합니다. Airnode가 완전 서버리스로 동작하기 때문에, 단일 클라우드 제공자의 가용 영역을 넘나들며 혹은 여러 클라우드 제공자 간에 쉽게 이중화가 가능합니다. 또한 Airnode의 컨테이너화된 버전을 온프레미스(on-premises) 환경에서 운영하는 것도 가능하지만, 거의 모든 사용 사례에서는 서버리스 버전을 사용하는 것이 권장됩니다.

Airnode는 MIT 라이선스로 오픈소스화되어 있으며, 퍼스트 파티 오라클을 실현할 준비가 되어 있습니다. API3는 보조금(grants)을 통해 Airnode 개발을 지원하며, 현재는 핵심 기술팀이 이를 유지관리하고 있습니다.

4.4 Airnode 프로토콜

오라클 문제 대신 더 명확하게 정의된 API 연결 문제를 선호하는 것과 마찬가지로, 오라클 노드는 모든 상상 가능한 목적에 사용할 수 있는 샌드박스가 아니라, API를 스마트 컨트랙트 플랫폼에 원활히 연동할 수 있도록 설계되어야 한다고 판단합니다. 이러한 철학에 따라, Airnode 프로토콜은 API가 스스로 발전시켜 온 패턴을 따르며 API-스마트 컨트랙트 플랫폼 간 인터페이스를 가능한 한 투명하고 마찰 없는 방식으로 구현하도록 설계되었습니다.

가장 먼저, 그리고 가장 널리 사용되는 API 스타일은 요청-응답(request-response) 패턴입니다. 사용자가 매개변수를 포함한 요청을 하면 API가 가능한 한 빨리 응답을 반환하는 방식입니다. Airnode가 첫 번째로 지원할 패턴이 바로 이 방식이며, 이는 동일한 패턴을 따르는 기존 API와 쉽게 표준화통합할 수 있습니다. 예시로는 특정 경기 결과를 요청하여 이를 전달받는 경우가 있으며, 이는 해당 예측 시장을 해결하는 데 활용될 수 있습니다. 또한 Airnode는 퍼블리시-서브스크라이브(publish-subscribe) 패턴도 지원할 예정입니다. 이는 사용자가 특정 매개변수 조건이 충족될 경우 오라클이 지정된 메서드를 호출하도록 요청하는 방식입니다. 예를 들어, 탈중앙화 거래소가 ETH 가격이 \$4000 이하로 하락했을 때 레버리지 포지션을 보유한 사용자의 청산 이벤트를 트리거하도록 오라클에 요청할 수 있습니다. 두 패턴 모두 현재 DeFi 애플리케이션에서 사용하는 실시간 데이터 피드 [2] 구현에 활용될 수 있으며, dAPI 형태로 훨씬 더 다양한 활용 사례를 지원할 수 있습니다.

4.3절에서 언급했듯이, Airnode 프로토콜은 요청자가 모든 가스 비용을 부담하도록 설계되었습니다. 여기에는 요청 실행 트랜잭션까지 포함됩니다. 이는 각 Airnode가 요청자마다 별도의 지갑을 보유하는 방식으로 구현되며, 이는 암호화폐 거래소가 사용자가 입금할 지갑을 자동으로 지정하는 방식과 유사합니다. 요청자는 네이티브 토큰(예: ETH)으로 해당 지갑에 일괄 충전하거나 요청별 소액 결제를 통해 자금을 공급할 수 있습니다. 이 지갑의 자금은 해당 요청자가 발생시키는 모든 요청을 처리하는 데 사용됩니다. 이 방식은 다음과 같은 중요한 장점을 제공합니다:

- 가스 비용과 결제 토큰 가격(예: LINK)의 변동성으로 인해 오라클이 경쟁력 있는 동시에 수익성 있는 가격을 책정하는 것은 사실상 불가능합니다. 온체인에서 가격을 동적으로 계산하려면 다수의 데이터 피드가 필요하고 요청당 상당한 가스 오버헤드가 추가됩니다. Airnode 프로토콜에서는 API 제공자가 가스 비용을 신경 쓸 필요가 없으며, 월 구독료와 같은 일반적인 가격 모델을 적용할 수 있습니다.
- 4.2절에서 언급했듯이, API 제공자가 일상 운영의 일부로서 법정화폐를 암호화폐로 전환해 노드 지갑에 충전하는 것은 비현실적입니다. 이 방식에서는 노드 운영자가 자신의 지갑 잔고를 고려할 필요가 전혀 없습니다.
- 최근 체인링크 데이터 피드에서 발생한 공격 사례 [27]에서 볼 수 있듯이, 공용 지갑을 사용해 요청을 처리하는 오라클 노드는 공격자가 스팸 요청을 보내 지갑을 탈취하는 공격에 취약합니다. 이에 대한 기존 해결책은 노드 운영자가 신뢰할 수 있는 주소 목록(화이트리스트)을 유지하는 것이지만, 어떤 컨트랙트를 신뢰할지 판단하기 어려울 뿐 아니라, 이는 사실상 공개 요청 서비스의 구현을 불가능하게 만듭니다. 이는 독립 생태계의 성장을 근본적으로 가로막는 중대한 문제입니다. Airnode는 요청자별 지갑이 해당 요청자의 요청에만 사용되므로, 이와 같은 공격에 취약하지 않습니다.
- 기존 오라클 노드는 낮은 가스 가격으로 제출된 단일 트랜잭션 때문에 전체 트랜잭션 큐가 막히는 상황을 허용할 수 없으므로, 모든 요청을 매우 높은 가스 가격으로 처리해야 합니다. 반면 Airnode 프로토콜에서는 요청자마다 별도의 트랜잭션 큐를 갖기 때문에 이러한 문제가 발생하지 않습니다. 따라서 긴 급하지 않은 요청의 경우, 요청자는 가스 가격을 매개변수로 지정하여 훨씬 낮은 비용으로 서비스를 이용할 수 있습니다. 이 방식은 특히 EIP-1559 [28]와 강력한 시너지를 발휘합니다.

마지막으로, Airnode 프로토콜의 수익화 접근 방식을 간단히 언급하겠습니다. 많은 프로젝트에서 특정 토큰을 프로토콜 핵심에 강제로 통합하여 해당 토큰의 필요성을 보장하려 시도하지만, 이는 상당한 가스 비용 오버헤드를 유발하고, 대체적인 수익화 방안을 심각하게 제한하며, 전반적인 마찰을 초래합니다. 이러한 이유로 Airnode 프로토콜은 의도적으로 전용 토큰을 사용하지 않습니다. 대신, 노드 운영자가 오라클 엔드포인트에 커스텀 인가(authorizer) 컨트랙트를 연결할 수 있도록 하며, 이를 통해 요청자에게 응답할지 여부를 온체인에서 구현 가능한 임의의 기준에 따라 결정할 수 있습니다. 인가 컨트랙트는 화이트리스트, 블랙리

스트, 월 구독료, 호출당 요금 부과 등을 강제하는 데 사용할 수 있습니다. 이 방식은 매우 유연할 뿐 아니라 가스 비용 오버헤드를 추가하지 않도록 구현됩니다. dAPI 수익화는 완전히 별개의 문제이지만, Airnode가 제공하는 유연성은 그대로 이어져, 예를 들어 사용자가 모든 가스 비용을 부담하는 dAPI를 구현할 수도 있으며, 이는 기존 오라클 솔루션으로는 불가능한 것입니다.

4.5 API 통합

API를 오라클에 통합하는 과정에는 흔히 ‘닭과 달걀 문제’가 존재합니다. 오라클 생태계에서 특정 API에 대한 수요가 존재하지 않으면, 그 API를 통합할 유인이 사라집니다. 반대로 API가 통합되지 않아 사용 불가능하다면, 그 API에 대한 수요를 창출할 애플리케이션이 개발되지 않습니다. 이는 체인링크(Chainlink) 생태계의 핵심 마찰 지점으로 확인되었으며, 이에 대한 해결책으로 Honeycomb API Marketplace가 제안되었습니다 [19]. Honeycomb은 많은 프리미엄 API를 체인링크 오라클에 통합했으며, 그 결과 당시 어떤 오라클 생태계에서도 볼 수 없던 다양한 API를 제공할 수 있었습니다.

Honeycomb은 범용 외부 어댑터(universal external adapter)와 새로운 방식을 활용하여 API를 체인링크 오라클에 선언적 방식으로 통합하였으며, 이를 위해 별도의 코드를 작성할 필요가 없었습니다. 이는 각 API 연산마다 외부 어댑터를 개발하는 방식 [29] 보다 우수한데, 통합 속도가 더 빠르고 오류 가능성이 적으며 비전문가도 쉽게 작업할 수 있기 때문입니다. 이러한 독자적 도구를 활용해 Honeycomb은 불과 몇 달 만에 수백 개의 고유 API 연산을 통합할 수 있었으며, 이는 당시 경쟁자들과 비교해 압도적인 격차였습니다.

API3가 그 잠재력을 충분히 실현하기 위해서는 수백, 더 나아가 수천 개의 퍼스트 파티 오라클이 필요합니다. 이를 통해 새로운 dAPI를 손쉽게 구축하거나 기존 dAPI를 재구성할 수 있기 때문입니다. 이를 달성하려면 API를 Airnode에 더욱 확장성 높은 방식으로 통합할 수 있어야 합니다. 이러한 목적을 위해, 앞서 설명한 독자적 통합 도구의 개선된 버전이 Airnode를 위해 오픈소스로 공개되었습니다. OpenAPI Specification [30] 형식을 차용한 Oracle Integration Specifications(OIS)은 API의 연산, 오라클의 엔드포인트, 그리고 양자 간의 매핑 방식을 정의합니다. Airnode 사용자는 단순히 해당 API의 OIS를 노드에 제공하는 것만으로 API를 오라클을 통해 서비스할 수 있습니다. 이 표준화된 형식으로 작성된 통합은 수집, 버전 관리, 배포가 매우 용이합니다.

OIS는 Airnode가 사용할 통합 사양을 설명하기 위해 설계된 JSON 파일입니다. 따라서 사람이 직접 읽기 편하도록 만들어진 것은 아니며, 수동으로 OIS를 작성해 통합을 지정하는 것은 쉽지 않습니다. 이 문제는 ChainAPI라는 통합 플랫폼을 통해 해결됩니다. ChainAPI는 사용자가 직관적인 그래픽 인터페이스를 활용해 자신들의 API에 대한 OIS를 생성할 수 있도록 합니다. 더불어 노드 대시보드, 엔드포인트 마켓플레이스 등 Airnode 사용자의 편의성을 높이는 추가 기능도 제공될 예정입니다. 결과적으로 API3는 dAPI를 구성할 수 있는 광범위한 퍼스트 파티 오라클 선택지를 확보하게 되며, 생태계 성장은 더 이상 통합 역량의 한계에 의해 병목되지 않게 됩니다.

5. 토크노믹스를 통한 거버넌스 탈중앙화 (Decentralizing Governance through Tokenomics)

단일 장애 지점(single point of failure)은 시스템의 일부가 실패할 경우 이를 보완할 중복성이 존재하지 않아 전체 시스템이 붕괴되는 치명적 요소입니다. 중앙화는 이러한 단일 장애 지점을 만들어내며, 탈중앙화는 이를 제거하는 것을 목표로 합니다. 블록체인 기반 애플리케이션은 탈중앙화를 내세우지만, 실제로는 여전히 중앙화된 요소를 포함하고 있는 경우가 많으며, 특히 거버넌스에서 두드러집니다 [31]. 본 절에서는 중앙화된 거버넌스로 인해 발생하는 문제와, API3가 잘 설계된 토크노믹스를 바탕으로 한 탈중앙화 자율조직(DAO, Decentralized Autonomous Organization) [32]을 통해 이를 어떻게 해결하는지 논의하겠습니다.

5.1 중앙화된 오라클 네트워크 거버넌스

탈중앙화 오라클 네트워크가 중앙화된 주체에 의해 구성·조정될 수 있다면, 그 거버넌스는 사실상 중앙화된 것입니다. 이는 거버넌스 실수가 눈에 띄지 않게 되어, 기본적인 API와 오라클이 올바르게 작동함에도 데이터 피드가 잘못된 정보를 보고하는 상황을 초래할 수 있습니다. 예를 들어, 체인링크(Chainlink) [15]의 은 가격 데이터 피드가 한동안 금 가격을 보고한 적이 있는데, 이는 인적 오류로 인한 거버넌스 실수 때문이었습니다 [33]. 당시 파생상품을 위한 탈중앙화 거래소인 Synthetix [34]는 해당 데이터 피드를 사용하고 있었으며, 일부 사용자가 이 오류를 악용해 이익을 얻었습니다 [35]. 중앙화된 거버넌스는 그 본질적인 불투명성으로 인해 열악한 관행이 사용되도록 허용하며, 이는 필연적으로 이러한 결과를 초래합니다. 그러나 이 사건이 보여준 더 중요한 문제는, 중앙화된 거버넌스 주체가 권한을 악의적으로 남용해 데이터를 쉽게 조작할 수 있다는 점입니다.

거버넌스 주체는 데이터 피드를 재구성할 권한을 가지고 있으며, 이는 오라클과 그 데이터 소스를 교체하는 것을 의미합니다. 장기적인 데이터 피드 유지 관리에는 필요하지만, 이는 동시에 데이터 피드 사용자가 거버넌스 주체의 다양한 남용 및 공격에 노출됨을 의미합니다. 따라서 사용자는 중앙화된 거버넌스 주체를 신뢰해야 하거나, 데이터 피드의 거버넌스를 탈중앙화하고 보안을 우선하는 인센티브 구조를 마련해야 합니다.

사용자가 중앙화된 거버넌스 주체를 완전히 신뢰할 수 있다고 판단한다면, 탈중앙화 오라클 네트워크를 사용하는 것은 비합리적이며, 해당 주체가 운영하는 중앙화된 오라클을 사용하는 것이 더 낫습니다. 첫째, 3.1절에서 논의했듯이, 중앙화된 오라클은 제3자 오라클이라는 공격 벡터가 없으므로 더 안전합니다. 둘째, 중앙화된 오라클은 다수의 오라클 노드 운영자를 조정하는 데서 발생하는 어려움이 없으므로, 가용성 측면에서 훨씬 뛰어난 성능을 제공합니다. 실제로 이러한 조정 실패는 데이터 피드 전체의 중단을 초래하기도 합니다 [36]. 마지막으로, 중앙화 오라클의 운영 비용은 오라클 네트워크보다 훨씬 낮습니다. 따라서 오라클 네트워크의 중앙화된 거버넌스가 정당화될 수 있는 경우는 없습니다.

5.2 자금 관리

초기 코인 공개(ICO)는 블록체인 프로젝트의 인기 있는 자금 조달 방식이었으며, 일반적으로 개발팀이 자금 전액을 전적으로 신뢰받아 관리하는 구조였습니다. 표면적으로는 합리적일 수 있으나, 토큰 가격이 투기적으로 상승할 경우 개발팀은 처음 투자자가 맡긴 것보다 훨씬 큰 금액을 통제하게 됩니다. 중앙화된 거버넌스가 부패와 밀접하게 연관되어 있음은 잘 알려져 있으며 [37], 이는 결국 엑시트 스캠부터 토큰 가격 조작을 위한 개발 자금 남용에 이르기까지 다양한 부정적 결과로 이어질 잠재력이 있습니다. 불투명한 예산 운영은 이러한 위험을 더욱 심화시키는데, 불행히도 이는 일반적인 관행입니다. 일부 프로젝트는 기술 개발 자금 외에 생태계 개발 자금을 별도로 두기도 하는데, 이 경우 개발팀이 생태계 전체를 대표하지 않으면서 자금을 통제하는 것은 더욱 정당화하기 어렵습니다.

이 문제의 해결책으로 DAO와 ICO를 결합한 DAICO 모델이 제안되었습니다. 이는 투자자들로 구성된 DAO가 개발팀에 대한 급여(stipend)를 배정하고, 이를 규제하거나 심지어 완전히 중단시킬 수 있는 구조입니다 [38]. 오늘날 DAO에서 성공적으로 채택된 더 유연한 접근 방식은 전체 개발을 보조금(grant) 방식으로 수행하는 것입니다 [39]. 이 방식에서는 DAO에 고정된 개발팀이 존재하지 않고, DAO가 개별 작업을 발주하여 제3자를 고용하는 구조입니다. 이는 실제 시장 가격에 기반한 정직하고 효율적인 개발 및 생태계 자금 배분을 가능하게 합니다.

5.3 API3 DAO

dAPI와 프로젝트 전체의 거버넌스를 탈중앙화하기 위해 API3는 DAO에 의해 운영됩니다. 거버넌스는 완전히 탈중앙화되고 개방적이며, 모든 이해관계자가 직접적으로 프로젝트 거버넌스에 참여할 수 있습니다. 이는 5.4절에서 설명된 메커니즘에 따라 API3 토큰을 통해 API3 DAO 내에서 투표권을 부여하는 방식으로 구현됩니다.

DAO는 스테이킹 인센티브와 담보 설정과 같은 메커니즘에 관한 고수준의 매개변수에 대해 투표합니다. 또한 DAO는 보조금을 집행하여 프로젝트의 전반적인 방향을 결정합니다. 더 세부적인 과제는 확장성 있는 거버넌스를 위해 계층적 팀 구조를 통해 수행됩니다.

예상되는 워크플로우는 사람들이 오프체인 팀을 구성해 일회성 프로젝트나 API3에 도움이 되는 지속적인 작업을 수행하기 위한 보조금을 신청하는 것입니다. 팀은 다중 서명 지갑(예: Gnosis Safe [40])을 통해 보조금을 신청하고, 제안이 승인되면 DAO는 해당 자금을 팀의 다중 서명 지갑으로 지급합니다. 또한 DAO는 과제에 따라 특정 트랜잭션을 실행할 권한을 해당 팀의 다중 서명 지갑에 부여할 수도 있습니다. 예를 들어, 특정 사용자에게 대한 dAPI 구독 요금을 설정하는 권한이 해당됩니다. 중요한 책임이나 대규모 예산이 수반되는 프로젝트의 경우, 팀 구성원은 자격 검증 및 시빌 공격 방지를 위해 신원을 공개해야 할 수도 있습니다.

기술 관련 보조금 주제의 예시는 다음과 같습니다:

- Airnode, dAPI 컨트랙트, API3 DAO 컨트랙트의 기술 개발
- API3 프론트엔드 개발(스테이킹, 서비스 보증 등)
- API3 생태계 프로젝트 개발
- 신규 API, dAPI 사용자, 스마트 컨트랙트 플랫폼 통합
- 특정 API 및 dAPI에 대한 정량적·정성적 위험 평가

- dAPI 관리
- 기사, 튜토리얼, 영상 등을 통한 개발자 홍보
- 기술 및 보안 감사
- 버그 바운티 프로그램, 해커톤 등 기획 및 운영

비기술적 과제 역시 보조금으로 수행됩니다:

- 신규 API 제공자, dAPI 사용자 발굴을 위한 사업 개발
- 특정 dAPI 사용자의 구독 및 서비스 보증 가격 책정
- 운영 및 재무 감사
- 결제 처리
- UI/UX 디자인
- 마케팅
- 법률 자문

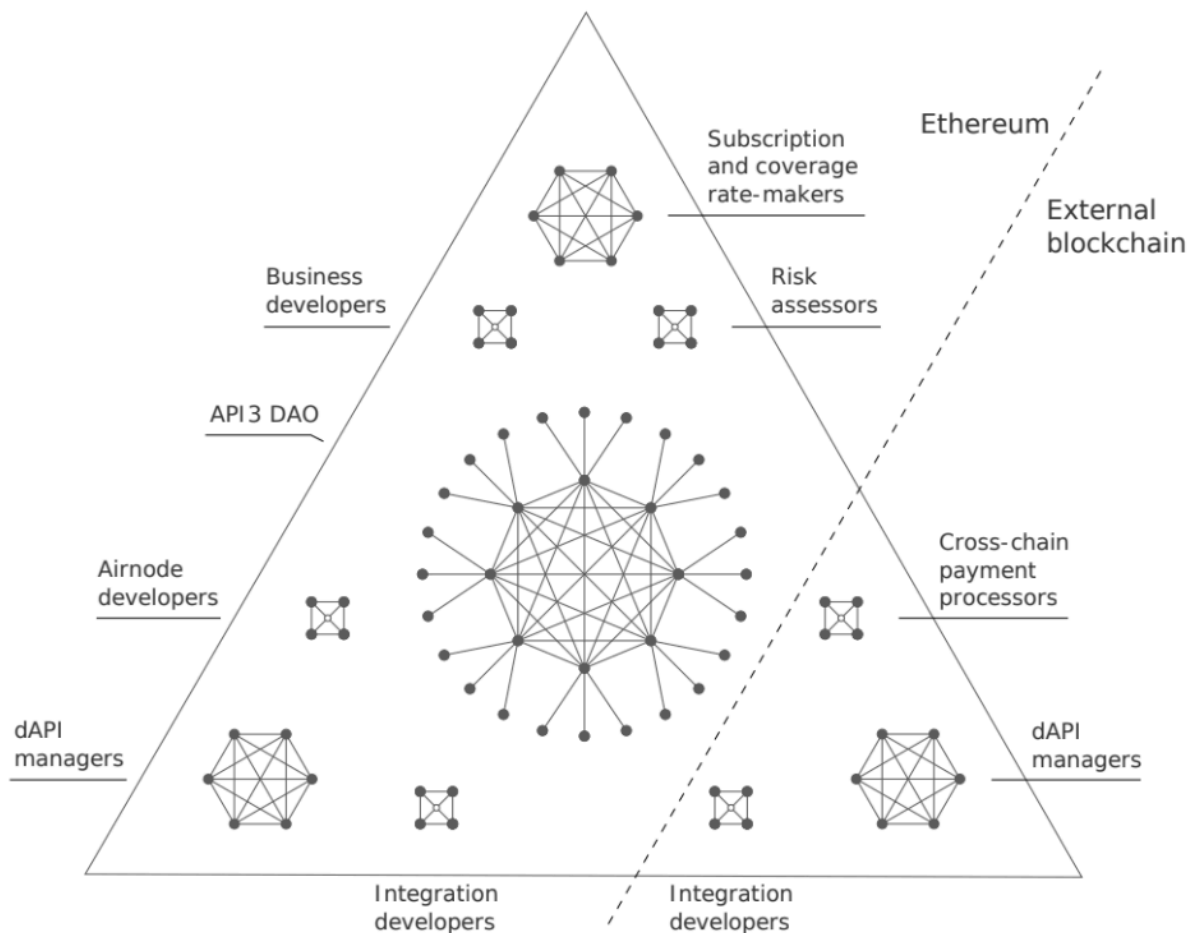


그림 6: 메인 DAO, 서브DAO, 체인 간에 분산된 팀으로 구성된 계층적 거버넌스 구조 예시. 메인 DAO는 자금을 선별적으로 배분하고 권한을 위임하여 거버넌스를 수행합니다. 팀이 더 이상 과제를 감당할 수 없는 규모에 도달하면, 해당 과제는 서브DAO에 배정됩니다.

이 팀 기반 거버넌스 구조는 DAO 수준에서 투표해야 하는 제안 수가 줄어들기 때문에 가스 비용 측면에서 확장 가능합니다. 또한 다양한 세부 사항에 모든 거버넌스 참여자가 지속적으로 관여할 필요가 없기 때문에 실제 운영 측면에서도 확장성이 높습니다. 더 나아가 dAPI 관리와 같은 중요한 작업이 전문가의 의견을 바탕으로 신속히 수행될 수 있도록 합니다. API3의 운영이 확장됨에 따라 이 거버넌스 계층은 추가적인 레이어를 필요로 할 수 있으며, 이는 서브DAO의 등장을 의미합니다(그림 6 참조).

이 구조가 효과적으로 작동하려면 DAO는 두 가지 원칙을 따라야 합니다. 첫째, 악의적이거나 무능한 팀이 초래할 수 있는 피해를 최소화하기 위해 팀의 권한은 가능한 한 제한되어야 하며, 이는 “최소 권한 원칙 (principle of least privilege)”으로 알려져 있습니다. 예를 들어, dAPI 관리 팀은 사용 중인 dAPI 전체를 완전히 재구성할 권한을 가질 수 없으며, 충분한 쿨다운 기간을 거쳐 개별 오라클만 교체할 수 있어야 합니다. 마찬가지로, 팀이 맡은 특정 책임을 수행하는 데 필요한 자금만을 지급하도록 마일스톤과 결과물이 활용되어야 합니다. 두 번째 원칙은 투명성입니다. DAO가 성과를 평가할 수 있으려면 팀은 DAO에 매우 상세하게 보고해야 합니다. 이러한 보고는 책임성을 강화할 뿐 아니라, dAPI 사용자와 일반 대중이 언제든지 API3 운영을 감사할 수 있도록 합니다.

5.4 API3 토크노믹스(API3 Tokenomics)

탈중앙화된 거버넌스에는 긍정적 결과와 부정적 결과를 모두 정확히 모델링할 수 있는 균형 잡힌 인센티브 메커니즘이 필요합니다. 다시 말해, 거버넌스 주체는 좋은 결과에 대해서는 보상을 받고, 나쁜 결과에 대해서는 처벌을 받아야 합니다. API3 토큰은 이를 실현하기 위해 세 가지 주요 기능을 수행하도록 설계되었습니다:

1. **스테이킹(Staking)**: 인플레이션 기반 보상을 제공하며, 이는 API3 서비스 이용 시 토큰 소각이나 타임락(time-lock)과 같은 디플레이션 메커니즘으로 균형을 맞춥니다.
2. **담보(Collateral)**: dAPI 오작동으로 발생하는 적격 피해에 대해 사용자를 보호하는 서비스 보증을 뒷받침합니다.
3. **거버넌스(Governance)**: API3 DAO에서 직접적인 대표권을 부여합니다.

스테이킹 기능은 API3 참여와 서비스 사용 확대에 기여하도록 재정적 인센티브를 제공합니다. 담보 기능은 참여자가 API3의 운영 리스크를 공유하게 하며, 리스크 최소화를 유도합니다. 마지막으로 거버넌스 기능은 이러한 인센티브를 실행에 옮길 수 있는 궁극적 수단을 부여합니다.

이 세 가지 기능이 반드시 일치해야 한다는 점이 중요합니다. 모든 거버넌스 주체는 서비스 사용을 극대화할 수 있도록 스테이킹 보상을 받아야 하며, 보안 리스크를 최소화할 수 있도록 자금을 담보로 제공해야 합니다. 이를 위해 API3는 단일 스테이킹 풀을 운영합니다. API3 토큰을 이 풀에 스테이킹하면 대표권과 스테이킹 보상을 얻을 수 있으며, 동시에 스테이킹된 토큰은 필요 시 유효한 서비스 보증 청구를 지급하기 위한 담보로 활용됩니다.

5.4.1 스테이킹

강력한 합의 메커니즘은 활성화된 기능에서 발생하는 가치를 참여자에게 보상하며, 이는 긍정적 피드백 루프를 형성합니다. 예를 들어, 이더리움은 무신뢰성 애플리케이션을 가능하게 하며, 이를 검증하는 채굴자는 사용자들이 지불한 트랜잭션 수수료 전체를 보상으로 받습니다. API3의 경우, 스테이커는 프로젝트를 지배하는 합의 메커니즘의 참여자이므로, dAPI 수익화의 주요 수혜자가 되어야 합니다. 이와 관련하여 EIP-1559

[28] 연구는 우리의 설계와 유사한 맥락을 갖습니다.

최근 런던 하드포크(London fork) [41] 이전까지, 이더리움 사용자는 트랜잭션을 블록에 포함시키기 위해 채굴자에게 직접 수수료를 지불했습니다. 그러나 EIP-1559는 이러한 수수료 기반 보상 체계가 경제적 불안정을 유발한다고 제시했습니다 [42]. 업그레이드 이후 사용자는 네트워크 사용량에 따라 변동하는 기본 수수료를 지불하며, 이 수수료는 소각됩니다. 채굴자는 전체 수수료 대신 안정적이고 인플레이션적인 블록 보상을 받게 되었습니다. (EIP-1559는 또한 ‘우선 수수료’ 메커니즘을 포함해, 이더리움 특유의 문제인 영클 블록(uncle block)이나 블록 크기 제한을 해결합니다. 이는 본 비유에 직접 대응하지는 않습니다.)

DAO가 수익을 스테이커에게 분배하는 구조는 EIP-1559 이전의 채굴 보상과 유사하게 불안정을 초래할 수 있습니다. 수익 분배 이벤트마다 인센티브가 불연속적으로 급등락하며, 이는 악용될 수 있습니다. 예를 들어, 일부 스테이커 그룹이 dAPI 구독료를 월별이 아닌 연 단위로 제안하고, 구독료가 지불되고 수익이 분배되자마자 언스테이킹을 해버리는 상황을 상정할 수 있습니다. 이는 장기 참여자를 불이익으로 몰아넣을 뿐만 아니라, 총 스테이킹 수량이 구독료 지급 주기에 따라 요동치게 만듭니다. 문제의 성격이 아날로지와 같듯, 해결책 역시 유사합니다. API3 DAO는 사용자가 API3 서비스를 이용하기 위해 API3 토큰을 소각하거나 타임락하도록 요구하며, 이는 스테이킹 보상 수혜자에게 간접적으로 이득을 주는 디플레이션 메커니즘을 형성합니다.

API3 서비스 사용이 증가하면 API3 토큰의 공급 부족을 초래하여, 스테이커뿐 아니라 모든 토큰 보유자에게 이익이 됩니다. 이는 스테이킹 보상이 적절히 규제되어 일정 비율의 토큰 보유자가 반드시 스테이킹하도록 유도해야 함을 의미합니다. 이를 위해 API3 DAO는 “스테이킹 목표(stake target)”를 설정합니다. 이는 전체 토큰 공급량 대비 비율로 정의되며, 매주 보상 금액은 반복적으로 조정됩니다. 즉, 스테이킹된 양이 목표치보다 낮으면 보상이 증가하고, 반대의 경우 감소합니다.

스테이킹 보상을 통해 양질의 거버넌스를 유도하는 것은 쉽지 않습니다. 거버넌스 품질은 쉽게 정량화할 수 없기 때문입니다. 예를 들어, 투표 참여는 일반적으로 바람직하게 여겨지지만, 이미 통과될 제안에 대한 투표는 불필요한 경우가 많으며, 의도적인 기권도 정당한 입장이 될 수 있습니다. 이에 대한 해결책으로 API3는 단기적 행위가 아닌 장기적 결과에만 보상을 제공합니다. 구체적으로, 스테이킹 보상은 매주 지급되지만, 각 보상은 1년이 지나야 출금 가능합니다. 이는 모든 스테이커가 1년 후 토큰 가격 극대화를 목표로 협력하도록 유도하며, 점진적 보상 해제 방식은 시스템 불안정을 방지합니다.

5.4.2 담보

만약 API3 스테이킹이 단순히 보상만 제공한다면, 거버넌스의 유일한 인센티브는 지급액을 극대화하는 것이 될 것입니다. 이는 dAPI 사용자의 수를 공격적으로 늘리고, 그에 따라 dAPI가 보장해야 하는 금액을 확대하는 방식으로 이루어질 것입니다. 그러나 3.2절에서 보았듯이, dAPI가 감당하는 총 부하가 커질수록 공격으로 인해 오작동할 가능성이 높아집니다. 따라서 이는 탈중앙화 데이터 피드에 있어 지속 가능한 거버넌스 전략이 아닙니다.

거버넌스 참여자를 우리가 피하려는 리스크에 노출시키는 것은 그들의 인센티브를 DAO와 일치시키는 효과를 가져옵니다. 따라서 dAPI 오작동이 발생할 경우, 거버넌스 참여자는 반드시 불이익을 받아야 합니다. 이를 한 단계 더 발전시켜, API3는 dAPI 사용자에게 적격 오작동(qualifying malfunction)에 대해 정량화 가능하고 무신뢰성 보안을 제공하는 온체인 서비스 보증 상품을 설계했습니다. 이 서비스 보증은 API3 토큰

큰 스테이킹 풀을 담보로 사용하며, 분쟁 해결 프로토콜을 통해 dAPI 오작동이 확인되면 사용자의 피해가 스테이킹 풀에서 보상됩니다. 이 서비스 보증의 구체적 구현 방식은 6장에서 다루겠습니다.

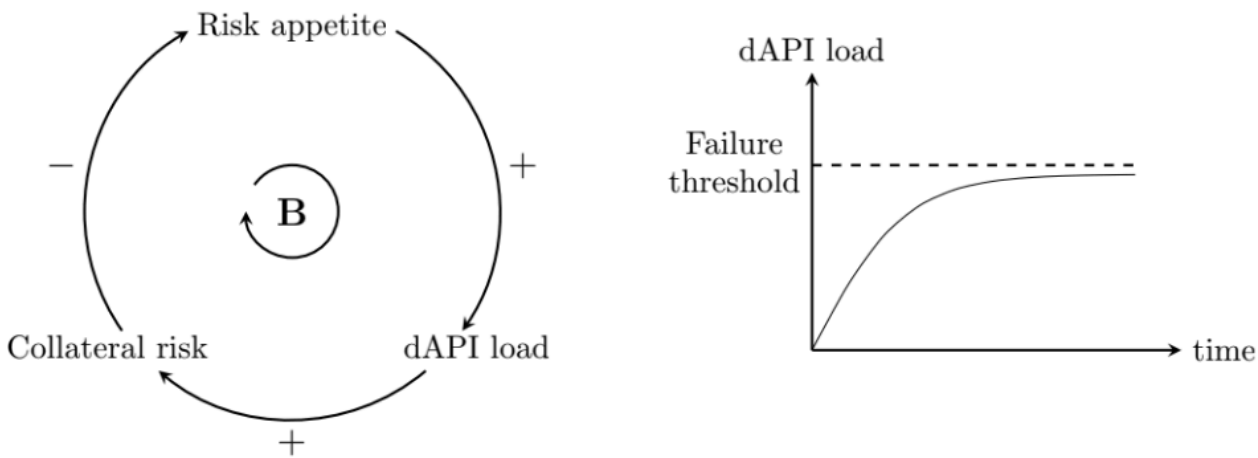


그림 7: API3 토큰의 스테이킹 및 서비스 보증 담보 기능은 균형 잡힌 거버넌스 인센티브를 형성합니다. (a) dAPI 사용자 수가 늘어나면 서비스 보증 청구 지급 가능성이 증가하여 부정적 피드백이 발생하고, 시스템 균형이 유지됩니다. B는 균형(balancing) 루프를 나타냅니다. (b) 시스템의 균형적 특성으로 인해 dAPI 부하는 무한정 증가하지 않고, DAO가 추정한 최대 부하 이하 수준에서 안정됩니다.

스테이킹 풀을 담보와 거버넌스에 동시에 사용하는 효과를 그림 7a의 시스템 다이어그램으로 살펴보겠습니다. DAO가 추가적인 리스크를 감수하고자 하면 새로운 dAPI 사용자를 온보딩하게 되고, 이는 dAPI 부하를 증가시킵니다. 그 결과 dAPI 오작동 확률이 증가하고, 서비스 보증 청구 지급 가능성이 높아지며, 전체 담보 리스크가 증가합니다. 담보 리스크가 증가하면 DAO의 리스크 감수 성향은 억제됩니다. 다시 말해, 서비스 보증으로 인한 부정적 피드백이 자기파괴적 성장을 방지하는 것입니다. 그림 7b는 이로 인해 나타날 것으로 예상되는 dAPI 부하 행동을 보여줍니다. DAO는 dAPI의 실패 임계값을 추정하고, 사용자를 온보딩하여 이 값에 수렴하도록 하지만 초과하지는 않습니다. 만약 DAO가 이 임계값을 과대평가했다면, dAPI가 오작동하고 거버넌스 참여자는 처벌을 받게 되며, 스테이킹한 자금이 피해 사용자에게 서비스 보증 청구 지급에 사용됩니다. 즉, 어느 경우든 dAPI 사용자는 보호됩니다.

5.4.3 거버넌스

API3 DAO에서 대표권을 얻는 유일한 방법은 API3 토큰을 서비스 보증 담보 풀에 스테이킹하는 것입니다. 따라서 거버넌스 참여자는 API3의 모든 리스크와 보상에 노출되며, 이를 최적화하기 위해 거버넌스를 수행합니다.

인플레이션 기반 보상과 담보로 활용되는 거버넌스 토큰 구조는 거버넌스 품질 측면에서 긍정적 피드백 루프를 형성합니다. 토큰 보유자는 인플레이션으로 인한 가치 손실을 피하려면 반드시 토큰을 스테이킹하고 리스크를 감수해야 합니다. 만약 거버넌스를 잘못 수행해 서비스 보증 청구로 인해 담보가 소진된다면, 해당 토큰은 시장에 다시 풀리게 되고 새로운 거버넌스 참여자에게 흡수됩니다. 반대로 초기 토큰 보유자가 거버넌스를 잘 수행해 시장에서 토큰 희소성을 만들어낸다면, 그들의 대표권 분포는 유지됩니다. 즉, 거버넌스 토큰을 담보로 사용하는 구조는 스스로 개선되고 실패로부터 회복할 수 있는 강력한 다윈주의적(Darwinian) 구조를

형성합니다.

6. 서비스 보증을 통한 정량적 보안

API3는 온체인 서비스 보증(on-chain service coverage)의 형태로 dAPI 사용자에게 정량화 가능한 수준의 보안을 제공합니다. 이를 통해 두 가지 목표가 달성됩니다: (1) 서비스 보증은 적격 dAPI 오작동이 발생할 경우 사용자에게 명확하고 무신뢰성 안전망 역할을 수행하며, (2) dAPI 오작동에 대해 거버넌스 참여자에게 책임을 부과함으로써 보다 안전한 dAPI 운영을 유도하는 인센티브를 형성합니다.

6.1 정량적 보안의 필요성

공학은 우리가 완전히 이해하지 못하는 재료를, 정확히 분석할 수 없는 형태로 모델링하여, 우리가 적절히 평가할 수 없는 힘을 견디도록 만드는 기술이다. 그리고 그 과정에서 대중이 우리의 무지를 의심할 이유가 없게 하는 것이다.

— Dr. A. R. Dykes

만약 엔지니어에게 “이 다리가 얼마나 많은 하중을 버틸 수 있습니까?”라고 물었을 때 “21개의 최고급 강철 보가 사용되었음을 보장할 수 있습니다”라는 대답을 듣는다면, 그 다리를 건너고 싶지 않을 것입니다. 최대 하중을 제시하지 못하는 것은 엔지니어링의 위험 신호이기 때문입니다. 3.2절에서 단일 오라클, 그리고 그에 따라 데이터 피드가 얼마나 많은 가치를 안전하게 지탱할 수 있는지를 설명하는 단순 모델을 제시한 바 있습니다. 비록 포괄적이지는 않지만, 이 모델은 탈중앙화 오라클 네트워크가 무제한의 가치를 안전하게 보장할 수 없음을 보여줍니다. 즉, 오라클이 안전하게 보장할 수 있는 한계가 존재해야 합니다. 다시 말해, 모든 블록체인 기술과 마찬가지로, 탈중앙화 오라클 네트워크는 무조건적으로 신뢰할 수 있는 것이 아니라 일정 범위 내에서만 신뢰되어야 합니다 [43]. 따라서 중앙화 [44]이든 탈중앙화 [15]이든 데이터 피드는 자신이 보장할 수 있는 한도를 정량적으로 제시할 책임이 있습니다.

이 문제에 대한 가장 잘 알려진 해결책 중 하나는 UMA 프로토콜 [45]에서 제안되었습니다. 제안된 방식은 게임 이론적 원리에 따라 데이터 피드가 보장할 수 있는 한도를 정량화할 수 있게 할 뿐 아니라, 그 한도를 정확히 설정할 수 있게 합니다. 저자들은 지나치게 안전한 데이터 피드는 사용자에게 불필요하게 비싸지므로, 최소한의 요구사항만 충족하도록 보안 수준을 설정하는 것이 비용 절감에 도움이 된다고 지적합니다. 그러나 그들은 자신들의 방식이 비용 효율성 측면에서 최적이라고 주장하는데, 이는 실제로는 크게 부정확합니다. 이 오류는 문제를 데이터 소스에 구매받지 않는 방식으로 접근했기 때문이며, 즉 오라클 문제를 풀려 한 것입니다. 제안된 해결책은 모든 오라클이 불신뢰 가능하다고 가정했을 때만 비용 효율적으로 보입니다. 이는 제3자 오라클에 대해서는 충분히 근접한 근사치일 수 있습니다. 반면 퍼스트 파티 오라클의 신뢰성은 훨씬 강력하게 활용될 수 있으며, API 제공자는 공격을 시도할 경우 잃을 것이 너무 많기 때문에, 매우 낮은 비용으로도 극도로 안전한 데이터 피드를 구축할 수 있습니다. 실제로, 평판 높은 단일 중앙화 거래소가 제공하는 데이터로 고가치 DeFi 상품이 안전하게 운영되는 사례가 이를 잘 보여줍니다 [44]. 따라서 입증된 신뢰성을 무시하고 비용 효율적인 해결책을 기대하는 것은 불가능합니다.

API3의 비전에 부합하는 이상적 해결책은 API 제공자의 신뢰성을 기반으로 정량적 보안을 제공하는 것이며, 이는 오프체인 정보를 통해서만 평가될 수 있습니다. 이를 위해 API3는 적격 오작동으로 인한 피해가 명확히 입증된 dAPI 사용자에게, 사전에 정해진 금액 한도 내에서 보상을 보장하는 서비스 보증을 제공합니다. 이는 사용자에게 더 유리한 해결책인데, 게임 이론적 대안은 인센티브가 잘못 모델링되거나 고려되지 않았을 경우 예기치 못하게 실패할 수 있기 때문입니다.

6.2 dAPI 서비스 보증(dAPI Service Coverage)

5.1절에서, 체인링크 데이터 피드가 Synthetix에 잘못 보고한 보안 사고를 언급했습니다. 당시 체인링크는 피해 규모를 4만 달러 미만으로 발표했고 [33], Synthetix는 이틀날 약 3만 6천 달러로 보고했습니다 [35]. 이후 Synthetix는 체인링크가 피해를 보상하겠다고 제안했고 이를 수락했습니다. 이 사건은 다음 사실들을 보여줍니다:

1. 적격 피해를 보상하는 서비스 보증은 데이터 피드 오작동에 대한 자연스럽고 명백한 해결책입니다.
2. 데이터 피드 오작동에 대한 책임은 일반적으로 거버넌스 주체에게 있다는 점이 널리 이해되고 있습니다.
3. 데이터 피드 오작동, 그 원인, 그리고 발생한 피해 규모는 며칠 내로 파악될 수 있습니다.

겉보기에는 이 사건이 별다른 문제 없이 해결되었습니다. 이는 피해 금액이 당사자들에게 비교적 사소했기 때문입니다. 그러나 대중이나 이해관계자 누구도 합의의 정확한 조건을 알 수 없었는데, 두 프로젝트 모두 중앙화된 방식으로 운영되었기 때문입니다. 그렇다면 피해 규모가 수십 배, 수백 배 컸다면 어떻게 되었을까요? 완전히 탈중앙화된 프로젝트는 이러한 사건을 어떻게 처리해야 할까요?

보험 사용은 거시경제 성장과 상관관계를 가질 뿐 아니라, 그 자체로 성장을 촉진하는 요인임이 입증되었습니다 [46]. 그럼에도 불구하고 블록체인 영역에서 보험은 극도로 과소 활용되고 있습니다. 그 주요 원인 중 하나는 보험이 본질적으로 제3자가 분쟁을 해결해야 한다는 점이며, 이는 탈중앙화 철학에 반하는 것입니다. 그러나 범용 온체인 분쟁 해결 프로토콜인 Kleros [5]의 등장은, 신뢰 불필요한 보험 유사 위험 보장 상품을 구축할 수 있도록 합니다.

API3는 Kleros와 공동으로, dAPI 사용자에게 정량적이고 신뢰 불필요한 보안을 제공하는 온체인 보험 유사 서비스 보증 상품을 개발할 것입니다. 이 서비스 보증은 특정 dAPI 오작동으로 인해 발생한 적격 피해를 사전에 정해진 한도까지 보상하도록 설계됩니다. 주목할 점은, API3가 이 서비스를 제공하지 않더라도 dAPI 사용자는 제3자 솔루션을 통해 온체인 보험이나 위험 보장 서비스를 받을 수 있다는 것입니다 [47]. 그러나 이러한 솔루션은 dAPI 리스크를 정확히 평가할 수 있는 정보와 전문성을 보유하지 못하기 때문에 매우 높은 보험료를 부과하게 될 것입니다. 또한 5.4.2절에서 설명했듯이, 제안된 서비스 보증은 API3 DAO 거버넌스 참여자가 스테이킹한 자금을 담보로 한다는 점에서 특별합니다. 따라서 이는 dAPI 사용자에게 보안을 제공할 뿐 아니라, dAPI가 최대한 안전하게 운영되도록 거버넌스를 유도하는 강력한 인센티브를 형성하여 서비스 보증 비용을 더욱 낮추게 됩니다.

6.3 서비스 보증 절차

사용자는 오프체인 채널을 통해 특정 dAPI에 구독을 요청하고, 해당 서비스에 대한 서비스 보증을 신청합니다. 보증 가능한 총액은 담보 풀의 크기에 의해 제한되며, DAO는 기존 보험 및 리스크 보장 지급능력(solvency) 관련 문헌 [48]을 참고하여 담보비율을 관리합니다. 관련 API3 팀은 dAPI 오작동 위험과 사용자의 특정 사용 사례를 조사하고, 서비스 보증 보험료를 산정하여 결제 관리 컨트랙트에 사용자별 요금을 기록합니다. 사용자가 컨트랙트에 요금을 지불하면, 해당 기간 동안 dAPI 접근 권한과 서비스 보증을 얻게 됩니다.

만약 dAPI 사용자가 오작동을 발견하면, 피해를 평가하고 적격 피해에 대한 보상을 청구하기 위해 온체인 서비스 보증 청구를 제출할 수 있습니다. 스테이킹 해제에는 충분한 리드타임이 설정되어 있어, 스테이커가 dAPI 오작동 발생 직후 청구 회피를 위해 서둘러 토큰을 인출하는 것을 방지합니다. 반대로 청구자는 남용을 막기 위해 일정 금액을 스테이킹해야 청구를 제기할 수 있습니다. API3 DAO는 (1) 청구 금액 전액을 지급하거나, (2) 합의 금액을 제안하여 청구를 직접 해결하거나, (3) 청구를 Kleros 법정으로 에스컬레이션하여 해당 금액이 지급될지를 판정받을 수 있습니다. 청구가 승인되면 토큰은 dAPI 사용자에게 이전되며, 이는 스테이커가 자신이 스테이킹한 비율에 따라 피해를 분담하는 구조를 의미합니다. 즉, 스테이킹 풀의 $p\%$ 를 보유한 사용자는 승인된 청구 금액의 $p\%$ 를 부담하게 됩니다.

위 구조는 모든 금액이 API3 토큰으로 표시된다는 가정에 기반합니다. 그러나 일부 사용자는 ETH와 같은 다른 암호화폐로 보상을 원할 수 있습니다. 이 경우, 단순히 유동성 공급자가 지급 시점에 ETH로 자동 변환하는 것만으로는 충분하지 않습니다. 왜냐하면 청구 시점과 지급 시점 사이에 API3/ETH 환율이 변동해 슬리피지가 발생하기 때문입니다. 이를 해결하기 위해 API3 DAO는 담보 풀과 동일한 지급능력 규정을 적용받는 추가 ETH 준비금을 유지하여, 가격 변동성을 흡수하고 사용자가 청구한 원래 금액이 지급되도록 보장할 수 있습니다.

6.4 위험 평가

데이터 피드가 제공할 수 있는 보안 수준을 정량화하는 것은 매우 어려운 문제입니다. 그러나 이 문제를 전통적 보험 및 리스크 보장 영역에 포함시킴으로써, 우리는 기존 보험 및 리스크 서비스에서 축적된 다양한 문헌과 전문성을 활용할 수 있습니다. 따라서 API3는 계리사, 통계학자, 데이터 과학자, 요율 산정자(rate-maker), 분석가, 법률 자문 등의 지원을 받아 정량적 보안을 제공하고 적격 청구를 적절히 매개변수화하는 어려운 과제를 수행할 수 있습니다.

위험 평가는 서비스 보증 가격 최적화와 지급능력 추정의 정확성을 위해 반드시 필요한 단계입니다. 여기에는 두 가지 주요 요소가 포함됩니다:

- 내부 요인: dAPI가 오작동할 가능성은 얼마나 되는가?
- 외부 요인: dAPI 오작동으로 인한 예상 피해 금액은 얼마인가?

내부 위험 요인 중 가장 중요한 것 중 하나는 오라클 수준의 실패이며, 이는 개별 API를 정성적으로 조사하고, 그 성능을 통계적으로 분석함으로써 추정할 수 있습니다. 예를 들어, 정성적 조사를 통해 특정 API 제공자가 5년 이상 영업을 지속해왔음을 확인한다면, 이는 심각한 시빌 공격 위험을 의미하지 않음을 알 수 있습니다. 반면, 통계적 분석에서 특정 API 제공자가 제공하는 데이터가 자주 합의값과 벗어난다면, 높은 정확성을 요구하는 데이터 소비자에게 문제를 야기할 수 있습니다. 이러한 평가 결과는 dAPI 설계 시 API 제공자

의 수와 선택 기준에 대한 지침으로도 작용할 수 있습니다. 예컨대 과부하된 dAPI에 더 많은 API를 추가하는 것이 서비스 보증 위험을 줄여 결과적으로 비용을 절감할 수 있음이 밝혀질 수 있습니다.

운영 리스크 또한 중요한 요소로, 운영 절차를 검토하는 감사를 통해 평가할 수 있습니다. 중요한 점은 이러한 연구가 API3 팀에 의해 수행되고 DAO에 공개적으로 보고된다는 사실로, 이는 사용자에게 타의 추종을 불허하는 투명성과 보안 보장을 제공합니다.

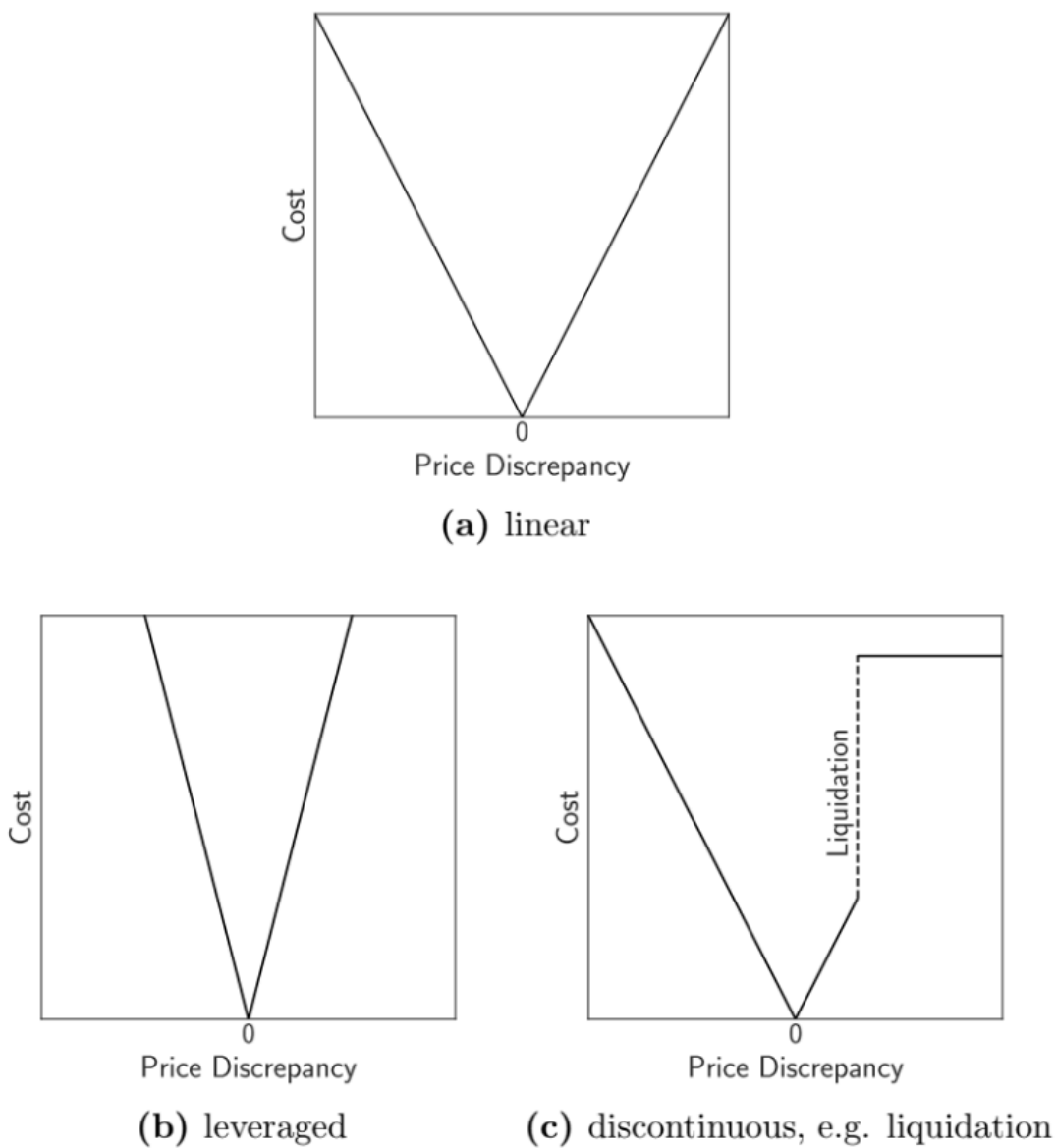


그림 8: dAPI에서 제공되는 데이터는 다양한 시스템에서 활용될 수 있으며, 이들 시스템의 메커니즘에서 선행성과 연속성이 어떻게 정의되는지가 입력 데이터 오류에 대한 민감도를 결정합니다.

외부 위험 요인은 오작동 발생 시 예상 피해 가치를 결정하며, 데이터가 어떻게 사용되는지가 중요한 요소입니다. 그림 8은 다양한 가상의 DeFi 애플리케이션에서 데이터 오류 비용을 예시적으로 보여줍니다.

그림 8a에서는 dAPI가 일반 거래소에 가격 데이터를 제공하며, 데이터 오류는 거래 당사자에게 선형적으로 비례하는 손익을 발생시킵니다(손실만 고려합니다). 이를 그림 8b와 비교해보면, 유사한 거래량을 가지지만 레버리지 포지션을 지원하는 거래소의 경우, 단순한 잘못된 보고도 훨씬 더 큰 피해를 야기할 수 있습니다. 마지막으로 그림 8c에서는 사용자가 숏 포지션을 보유한 경우를 보여줍니다. 보고된 데이터가 약간만 상향 왜곡되어도 강제 청산되어 불균형적으로 큰 손실을 초래할 수 있습니다. 이러한 예시는 특정 사용자의 서비스 보증 위험을 평가할 때, 해당 사용자가 dAPI 데이터를 정확히 어떻게 사용할 것인지를 고려하지 않고서는 불가능함을 보여줍니다.

서비스 보증에는 보다 정성적인 측면도 존재합니다. 구체적으로, API3와 dAPI 사용자는 서비스 보증 정책을 합의하며, 이는 dAPI 오작동의 정의, 유효한 청구 요건, 피해 산정 방식과 같은 기준을 포함합니다. Kleros 배심원은 서비스 보증 청구 지급 여부를 판정할 때 이 정책과 용어를 참조하게 됩니다. 이 조항들은 서비스 보증 효율 산정에 직접적으로 영향을 미칩니다. 예를 들어, 모든 오작동을 보장하는 서비스 보증은 단순히 다운타임만 보장하는 서비스 보증보다 비용이 더 높게 책정됩니다.

6.5 확장 솔루션과 서비스 보증

DeFi와 같은 고가치(High-value) 사용 사례가 대중화되면서 이더리움 네트워크는 혼잡해지고 있습니다. 이는 트랜잭션 수수료를 증가시키며 데이터 피드 운영 비용에도 직접적인 영향을 미칩니다. 따라서 dAPI 서비스를 합리적인 비용으로 제공하기 위해 확장 솔루션을 활용하는 것이 중요해집니다.

기존의 탈중앙화 오라클 솔루션들은 오프체인 확장 솔루션을 사용할 것을 제안합니다 [15, 16]. 그러나 이러한 솔루션에는 사용자가 명확히 평가하기 어려운 불투명한 보안적 함의가 뒤따릅니다. 첫째, 확장 솔루션은 일반적으로 보안 보장이 완화된 경향이 있으며, 사용자가 그 결과를 충분히 이해할 것이라고 기대하는 것은 비현실적입니다. 둘째, 맞춤형 암호 함수 구현의 보안 문제나, 2계층 솔루션의 서비스 거부와 같은 추가적인 운영 리스크도 존재합니다. 따라서 사용자가 확장 솔루션에 의존하는 데이터 피드 사용을 꺼리는 것은 당연한 일입니다.

dAPI 서비스 보증은 그 유연성 덕분에 이러한 문제의 의외의 해결책이 될 수 있습니다. API3 DAO가 특정 사용 사례에 대해 어떤 확장 솔루션이 충분히 신뢰할 만하다고 판단할 경우, 해당 dAPI는 그 솔루션을 활용할 수 있으며, 서비스 보증은 해당 확장 솔루션으로 인해 발생할 수 있는 피해까지 보장할 수 있습니다. 이 경우에도 서비스 보증 청구 절차는 동일하게 작동하며, 궁극적으로 중요한 것은 서비스가 dAPI 사용자에게 올바르게 전달되었는지 여부입니다.

7. 결론

API3는 탈중앙화 애플리케이션을 기존 Web API가 제공하는 풍부한 데이터와 서비스와 연결하여, 탈중앙화를 훼손하지 않고 블록체인의 적용 범위를 확장합니다. 이는 API3 DAO에 의해 대규모로 설정·운영·수익화되는 완전한 탈중앙화 및 블록체인 네이티브 API, 즉 dAPI를 통해 이루어질 것입니다.

API3 솔루션은 설계상 다양한 특성을 갖추고 있습니다. 그중 가장 중요한 것은 보안입니다. dAPI는 대안적 솔루션에서 상존하는 중대한 리스크 요인인 제3자 오라클에 의존하지 않습니다. 또한 dAPI 서비스 보장은 사용자에게 정량화 가능하고 신뢰 불필요한 보안을 제공하여, 탈중앙화 애플리케이션이 API 서비스를 수신하는 가장 안전한 방법으로서 API3의 위치를 확고히 합니다.

두 번째 특성은 강인성입니다. Airnode는 서버리스(serverless) 기술을 사용하여 다운타임에 매우 강합니다. 또한 상태 비저장(stateless) 노드 설계는 버그나 불리한 네트워크 조건에도 쉽게 영향을 받지 않으며, API3 오라클은 강인성을 염두에 두고 설계되었습니다. 더 나아가 dAPI는 잘 설계된 인센티브를 통해 리스크와 보상의 자율 균형을 유지하는 DAO에 의해 운영되므로, 견고한 위험 완화 프레임워크를 제공합니다.

세 번째 특성은 비용 효율성입니다. dAPI는 중개인을 제거하여 ‘중개인 세금’을 지불할 필요가 없습니다. 또한 퍼스트 파티 오라클로 구성된 데이터 피드는 오라클 수준에서 과도한 중복성을 요구하지 않으므로, 동일한 수준의 탈중앙화를 더 적은 오라클로 달성하며 가스 비용을 크게 절감합니다.

마지막으로 API3 솔루션은 유연성을 달성합니다. 이는 실제 이해관계(stake)를 가진 참여자들에게 거버넌스를 완전히 탈중앙화함으로써 가능하며, 프로젝트는 본 백서에 제시된 내용에 국한되지 않고 새로운 도전과 요구에 맞추어 지속적으로 진화할 수 있습니다.

1세대 탈중앙화 애플리케이션은 블록체인 내부에 제한되어 있었습니다. 오늘날에는 오프체인 세계와 제한적·준탈중앙화 방식으로 상호작용할 수 있는 탈중앙화 애플리케이션이 존재합니다. API3는 오프체인 세계와 의미 있게 상호작용하며, API를 진정으로 탈중앙화되고 신뢰 최소화된 방식으로 활용하는 제3세대 탈중앙화 애플리케이션의 진화를 이끌 것입니다.

참고문헌

[1] DeFi Pulse. <https://defipulse.com/>.

[2] B. Liu and P. Szalachowski, “A first look into DeFi oracles,” *arXiv preprint arXiv:2005.04377*, 2020.

[3] CLCG, “Honeycomb API Marketplace.” <https://honeycomb.market/>.

[4] CLCG, “Honeycomb smart contract hackathon.” <https://honeycomb.devpost.com/>, 2019.

[5] C. Lesage, F. Ast, and W. George, “Kleros: Short paper,” *Whitepaper v1.0.7*, 2019. <https://kleros.io/assets/whitepaper.pdf>.

[6] G. Collins and D. Sisk, “API economy: From systems to business services,” *Deloitte Insights*, 2015.

[7] MuleSoft, “Connectivity benchmark report,” 2019.

- [8] B. Iyer and M. Subramaniam, "The strategic value of APIs," *Harvard Business Review*, 2015.
- [9] R. Narain, A. Merrill, and E. Lesser, "Evolution of the API economy: Adopting new business models to drive future innovation," *IBM Institute for Business Value*, 2016.
- [10] Capgemini, Efma, "World FinTech report 2019," 2019.
- [11] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Whitepaper*, 2009. <https://bitcoin.org/bitcoin.pdf>.
- [12] N. Szabo, "Smart contracts." 1994.
- [13] V. Buterin, "A next-generation smart contract and decentralized application platform," *Whitepaper*, 2014. <https://ethereum.org/en/whitepaper/>.
- [14] J. Peterson, J. Krug, M. Zoltu, A. K. Williams, and S. Alexander, "Augur: A decentralized oracle and prediction market platform," *Whitepaper v2.0*, 2019. <https://augur.net/whitepaper.pdf>.
- [15] S. Ellis, A. Juels, and S. Nazarov, "ChainLink: A decentralized oracle network," *Whitepaper v1.0*, 2017. <https://research.chain.link/whitepaper-v1.pdf>.
- [16] Band Protocol. <https://bandprotocol.com/>.
- [17] A. S. de Pedro, D. Levi, and L. I. Cuende, "Witnet: A decentralized oracle network protocol," *Whitepaper*, 2017. <https://arxiv.org/pdf/1711.09756.pdf>.
- [18] "Tellor: A decentralized oracle," *Whitepaper*. <https://docs.tellor.io/tellor/whitepaper/introduction>.
- [19] B. Benligiray, D. Connor, A. Tate, and H. Vanttinen, "Honeycomb: An ecosystem hub for decentralized oracle networks," *Whitepaper*, 2019. <https://raw.githubusercontent.com/clc-group/honeycomb-whitepaper/master/honeycomb%20whitepaper.pdf>.
- [20] Gnosis, "Omen and the next generation of prediction markets." <https://blog.gnosis.pm/omen-and-the-next-generation-of-prediction-markets-2e7a2dd604e>, 2020.
- [21] Capgemini, Efma, "World insurance report 2019," 2019.
- [22] J. R. Douceur, "The Sybil attack," in *Proc. Int. Workshop on Peer-to-Peer Systems*, pp. 251–260, 2002.
- [23] S. Frederick, G. Loewenstein, and T. O'donoghue, "Time discounting and time preference: A critical review," *Journal of Economic Literature*, vol. 40, no. 2, pp. 351–

401, 2002.

[24] Chainlink, “Price reference data.” <https://feeds.chain.link/>.

[25] Practical Law, “Data licensing: Taking into account data ownership and use.” <https://legal.thomsonreuters.com/en/insights/articles/data-licensing-taking-into-account-data-ownership>.

[26] Compound Finance, “Open price feed.” <https://compound.finance/prices>.

[27] The Block, “Chainlink nodes were targeted in an attack last weekend that cost them at least 700 ETH.” <https://www.theblockcrypto.com/post/76986/chainlink-nodes-attack-eth>, 2020.

[28] V. Buterin, E. Conner, R. Dudley, M. Slipper, and I. Norden, “EIP-1559: Fee market change for ETH 1.0 chain.” <https://eips.ethereum.org/EIPS/eip-1559>.

[29] Chainlink, “Chainlink external adapters.” <https://github.com/smartcontractkit/external-adapters-js>.

[30] OpenAPI Initiative, “The OpenAPI specification.” <https://github.com/OAI/OpenAPI-Specification/>.

[31] A. Walch, “Deconstructing ‘decentralization’: Exploring the core claim of crypto systems,” *Crypto Assets: Legal and Monetary Perspectives* (OUP, Forthcoming), 2019.

[32] V. Buterin, “DAOs, DACs, DAs and more: An incomplete terminology guide.” <https://blog.ethereum.org/2014/05/06/daos-dacs-das-and-more-an-incomplete-terminology-guide/>, 2014.

[33] Chainlink, “Improving and decentralizing Chainlink’s feature release and network upgrade process.” <https://blog.chain.link/improving-and-decentralizing-chainlinks-feature-release-and-network-upgrade-process/>, 2020.

[34] Synthetix, “Synthetix litepaper,” *Whitepaper v1.4*, 2020. <https://docs.synthetix.io/litepaper>.

[35] Synthetix, “Update on XAG pricing incident.” <https://blog.synthetix.io/update-on-xag-pricing-incident/>, 2020.

[36] DeFi Pulse, “DeFi status report post-Black Thursday.” <https://defipulse.com/blog/defi-status-report-black-thursday/>, 2020.

[37] R. Fisman and R. Gatti, “Decentralization and corruption: Evidence across countries,” *Journal of Public Economics*, vol. 83, no. 3, pp. 325–345, 2002.

[38] V. Buterin, “Explanation of DAICOs.” <https://ethresear.ch/t/explanation-of->

[daicos/465](#), 2018.

[39] “dxDAO: Toward super-scalable organizations,” *Whitepaper*, 2019.
https://github.com/gnosis/dx-daostack/blob/master/dxdao_whitepaper_v1.pdf.

[40] Gnosis, “Gnosis safe.” <https://gnosis-safe.io/>.

[41] Ethereum, “The history of Ethereum.”
<https://ethereum.org/en/history/#london>, 2021.

[42] T. Roughgarden, “Transaction fee mechanism design for the Ethereum blockchain: An economic analysis of EIP-1559,” *arXiv preprint arXiv:2012.00854*, 2020.

[43] P. De Filippi, M. Mannan, and W. Reijers, “Blockchain as a confidence machine: The problem of trust & challenges of governance,” *Technology in Society*, vol. 62, 2020.

[44] Coinbase, “Introducing the Coinbase price oracle.”
<https://blog.coinbase.com/introducing-the-coinbase-price-oracle-6d1ee22c7068>, 2020.

[45] UMA, “UMA data verification mechanism: Adding economic guarantees to blockchain oracles,” *White paper vo.2*, 2020.
<https://github.com/UMAprotocol/whitepaper/blob/master/UMA-DVM-oracle-whitepaper.pdf>.

[46] J. F. Outreville, “The relationship between insurance and economic development: 85 empirical papers for a review of the literature,” *Risk Management and Insurance Review*, vol. 16, no. 1, pp. 71–122, 2013.

[47] H. Karp and R. Melbardis, “Nexus Mutual: A peer-to-peer discretionary mutual on the Ethereum blockchain,” *Whitepaper*.
https://nexusmutual.io/assets/docs/nmx_white_paperv2.3.pdf.

[48] “Directive 2009/138/EC of the European Parliament and of the Council of 25 November 2009 on the taking-up and pursuit of the business of Insurance and Reinsurance (Solvency II) (recast).” <http://data.europa.eu/eli/dir/2009/138/2019-01-13>.

백서 버전

- **v1.0.3** – Erich Dylus, Burak Benligiray, June 2022.

- **v1.0.2** – Burak Benligiray, Derek Croote, November 2021.
- **v1.0.1** – Burak Benligiray, November 2020.
- **v1.0.0** – Burak Benligiray, Saša Milić, Heikki Vanttinen, September 2020.

용어집(Glossary)

Airnode: API 제공자가 직접 운영할 수 있도록 설계된 완전 서버리스(serverless) 오라클 노드.

API: 다른 애플리케이션이 프로그래밍 방식으로 상호작용할 수 있도록 하는 애플리케이션의 기술적 인터페이스. 외부 접근이 가능한 API(즉, Web API)는 기업이 데이터와 서비스를 수익화하는 데 활용될 수 있음.

API 제공자: API를 통해 데이터와 서비스를 수익화하는 기업.

API3: dAPI를 구축, 관리, 수익화하는 프로젝트.

API3 DAO: API3 프로젝트의 거버넌스 기구.

API3 토큰: API3 생태계의 인센티브를 정렬하는 데 사용되는 토큰. API3 DAO에서 의결권을 부여함.

ChainAPI: 제3자 API–Airnode 통합 플랫폼. <https://chainapi.com/> 에서 접근 가능. 통합 도구, 유틸리티, 퍼블릭 오라클 엔드포인트 마켓플레이스를 제공하며, Honeycomb API Marketplace의 정신적 후속작이라 할 수 있음.

DAO: 탈중앙화 자율조직(Decentralized Autonomous Organization). 온체인 조직의 거버넌스를 민주화하기 위해 사용되는 다중 사용자 스마트 컨트랙트.

dAPI: 탈중앙화 API. 퍼스트 파티 오라클로 구성된 데이터 피드를 의미하며, API3 DAO에 의해 탈중앙화 된 방식으로 거버넌스됨.

Honeycomb API Marketplace: API 중심의 마켓플레이스/리스트 서비스로, API3의 창립 멤버 일부가 체인링크 오라클을 위해 구축한 플랫폼.

오라클: 스마트 컨트랙트 플랫폼과 데이터를 송수신할 수 있는 에이전트. 예: 자산 가격 데이터를 블록체인에 기록하는 역할. 오라클은 노드와 요청자와 통신하기 위해 사용하는 프로토콜을 구현하는 스마트 컨트랙트를 포함함.

오라클 노드: 오라클의 오프체인 기능을 수행하는 애플리케이션. 예: API를 호출하여 자산 가격 데이터를 얻고 이를 블록체인에 기록함.

Web 3.0: 블록체인 기술 위에 구축된 탈중앙화 웹. (본 문서에서는 시맨틱 웹(Semantic Web)을 의미하는 용어로 사용되지 않음.)

Web API: 웹을 통해 접근할 수 있는 API. 즉, 사설 네트워크에서만 접근 가능한 것이 아니라 인터넷을 통해 접근 가능한 API.