인간 중심에서 에이전트 네이티브로: 자율형 AI를 가능하게 하는 트러스트리스 결제 인프라

1. 에이전틱 시대의 시작 (Entering the Agentic Future)

1.1 자율형 AI의 약속과 한계 (The Promise and Prison of Autonomous AI)

인공지능은 지금 전환점에 서 있습니다. GPT 모델은 매일 수십억 건의 요청을 처리하며 복잡한 추론 과제를 놀라운 정확도로 해결하고, Claude는 수천 개의 함수 호출을 조합해 정교한 다단계 워크플로우를 실행합니다. 이제 언어 모델은 복잡한 계획을 안정적으로 수행할 수 있으며, 그 능력은 이미 입증되었고 실사용 환경에 투입될 준비가 끝났습니다.

이 기술적 성숙은 인간 중심의 운영 구조를 대체하고 완전한 자율성과 신뢰 제거(Trustless)를 기반으로 한 에이전트 네이티브 아키텍처로의 구조적 전환을 예고합니다. 자율형 에이전트는 마이크로초 단위의 의사결정 속도와 분산형 인프라 최적화 능력을 통해 인간이 통제 가능한 수준을 넘어선 정밀도와 처리량을 달성하고 있습니다. 이들은 분산 시스템을 조율하고, 수십억 달러 규모의 포트폴리오를 관리하며 수많은 인간 운영팀조차 감당하지 못할 속도와 규모로 복잡한 결정을 내립니다. 맥킨지는 이러한 에이전트들이 2030년까지 연간 4.4조 달러의 가치를 창출할 것이라고 전망하며, 이는 현재의 기하급수적인 확산 속도를 고려할 때 오히려 보수적인 추정치입니다.

그러나 역설이 존재합니다. 에이전트를 가능하게 해야 할 인프라가 오히려 그들의 자율성을 억압하고 있습니다. 오늘날의 자율형 에이전트는 인간 개입을 전제로 설계된 중앙화 인프라 구조 속에서 여전히 제약받고 있습니다. 초단위로 글로벌 시장을 분석할 수 있는 온체인 에이전트가 전통 금융망을 통해 해외 결제를 처리하는 데 며칠을 소비하며, 핵심 의사결정을 수행하는 에이전트조차 자신이 정의된 규칙(Defined Constraints) 안에서 작동하고 있음을 암호학적으로 증명(Cryptographically Prove) 할 방법이 없습니다. 사용자는 불가능한 선택을 강요받습니다 — 에이전트의 자율적 결제 행위를 전적으로 신뢰하고 모든 리스크를 감수하거나, 모든 거래를 직접 승인해 에이전트의 본질적 가치를 스스로 훼손하거나.

이것은 지능(Intelligence)의 문제가 아닙니다. 지능의 발전은 이미 한계를 넘어섰지만, 에이전트가 자율적으로 작동할 신뢰 기반 인프라는 아직 존재하지 않습니다. 인프라 혁신이 더 이상 선택이 아닌 불가피하고 시급한 과제가 된 데에는 세 가지 근본적인 요인이 있습니다.

첫째, 모델의 준비도**(Model Readiness) —** 언어 모델은 이미 프로덕션 수준의 일관성과 복잡한 계획 실행 능력을 확보했습니다. 기술적 문제는 더 이상 논쟁의 대상이 아닙니다.

둘째, 기업의 수요**(Enterprise Demand) —** 기업은 에이전트에게 실질적인 권한을 부여해 효율을 극대화할 것인지, 아니면 권한을 제한해 경쟁력을 잃을 것인지라는 불가능한 선택 앞에 서 있습니다. 현재의 상태는 지속될 수 없습니다.

셋째, 규제의 현실**(Regulatory Reality) —** EU AI Act를 비롯한 글로벌 프레임워크는 알고리즘의 책임성과 에이전트 행위의 암호학적 증명**(Cryptographic Proof of Agent Behavior)** 을 요구합니다. 이제 컴플라이언스는 선택이 아닌 의무입니다.

모델은 이미 준비되었습니다**.** 기업은 한계에 다다랐고**,** 규제 기관은 지켜보고 있습니다**.**

Kite는 신뢰를 약속이 아닌 코드로 대체하며**,** 에이전트 인프라 혁명의 시작을 알립니다**.**

1.2 인간 중심 인프라가 자율형 에이전트를 한계에 가두는 이유

1.2.1 인증의 역설 **(The Authentication Paradox)**

현재 디지털 서비스는 '접근을 요청하는 주체는 인간이다' 라는 근본적 전제 위에서 설계되어 있습니다. 이 전제는 사용자 이름과 비밀번호 조합부터 생체인식 인증 시스템에 이르기까지, 인증 인프라의 모든 계층에 깊이 내재되어 있습니다. 그러나 이러한 인간 중심 설계는 AI 에이전트에게 세 가지 핵심 차원에서 드러나는 구조적 인증의 역설(Authentication Paradox) 을 야기합니다.

자격 증명 관리의 위기 **(The Credential Management Crisis)**

기업 환경에서 AI 에이전트를 실제로 배포할 때 자격 증명 관리의 복잡성은 기하급수적으로 증가합니다. 예를 들어 하나의 조직이 50개의 에이전트를 20개의 서비스에 연결할 경우, 총 1,000개의 고유한 자격 증명 관계를 관리해야 합니다. 각 관계는 별도의 프로비저닝, 회전 주기, 접근 정책, 감사 추적을 요구하며, M개의 에이전트가 N개의 서비스를 이용할 경우 M×N개의 자격 증명 관계가 생성됩니다. 각 관계는 서로 다른 보안 요건과 실패 지점을 갖습니다. 실제로 Claude MCP 통합 사례에서도, 몇 개의 에이전트만 배포하더라도 수십 개의 API 키를 관리해야 하는 상황이 이미 나타나고 있습니다.

자격 증명의 급증은 연쇄적인 취약점을 초래합니다. 현재 표준으로 널리 쓰이는 장기 API 키(서비스 계정)는 권한 범위가 지나치게 넓습니다. 각 에이전트와 서비스마다 세분화된 자격 증명을 적용하려면 복잡한 양방향 구성이 필요하기 때문입니다. 그 결과 키가 유출되면 공격자는 인간 사용자나 관리자에게 부여된 권한으로 장기간 접근할 수 있게 됩니다. 이처럼 보안 모델은 암호학적 제약 대신 '은폐에 의한 보안(security through obscurity)'에 의존하게 되며, 자격 증명이 노출되지 않기를 바라는 수준의 안이한 방어에 그치고 맙니다.

아이덴티티 검증의 불가능성 **(Identity Verification Impossibility)**

에이전트와 주체(Principal) 간의 암호학적 결합이 부재함으로써 근본적인 아이덴티티 위기가 발생합니다. 서비스는 “앨리스의 트레이딩 에이전트”가 실제로 앨리스의 소유인지, 혹은 이를 사칭하는 악의적 행위자인지 확인할 방법이 없습니다. 현재의 인증 체계는 합법적인 에이전트가 정의된 규칙 내에서 작동하는 경우, 침해된 에이전트가 위장하는 경우, 그리고 다수의 에이전트가 동일한 주체를 사칭하는 경우를 구분하지 못합니다.

이러한 아이덴티티 불명확성으로 인해 서비스는 결국 이분법적 선택을 강요받습니다.

자격 증명만으로 모든 접근을 허용하거나, 반대로 에이전트의 접근을 전면 차단하는 것입니다.

두 선택 모두 자율형 운영에 필수적인 **세분화된 조건 기반 권한 부여(Fine-Grained, Constraint-Based Authorization)**를 구현하지 못합니다.

OAuth의 한계 **–** 자율형·비용 연동형 에이전트에 대한 실패 **(OAuth Fails for Autonomous, Cost-Related Agents)**

기존의 인가 프레임워크인 OAuth는 인간이 감독하는 예측 가능한 API 호출을 전제로 설계되었습니다. 그러나 자율형 에이전트는 매분 수천 건의 비결정적(non-deterministic) 연산을 수행합니다. OAuth는 기술적으로 세분화된 접근 범위(Scope)를 지원하지만, 근본적인 한계는 한 번의 인가 후 맥락을 인식하지 못한다는 점입니다.

AI 에이전트가 OAuth 토큰을 획득한 이후, 시스템은 다음과 같은 핵심적인 런타임 대응하지 못합니다. 에이전트의 행동이 갑자기 변해 보안 침해를 시사하는가? 특정 행위가 연계된 서비스 전반에 걸쳐 고비용 연산을 유발하는가? 누적 지출이 합리적 한도를 초과했는가?

현재의 권한 부여 모델이 실패하는 이유는 다음과 같습니다.

- 경직된 평가**(Rigidly Evaluated):** 권한은 인증 시점에만 검사되고, 이후 변화된 맥락이나 비용에 따라 재검토되지 않습니다.
- 경제적 맹목**(Economically Blind):** 리소스 소비나 지출 속도를 추적하는 네이티브 메커니즘이 존재하지 않습니다.
- 조합적 폭발**(Compositionally Explosive):** 에이전트가 하위 에이전트에 권한을 위임하면 통제와 검증이 어려운 복잡한 권한 구조가 만들어집니다.

결과적으로, 각 통합 서비스는 잠재적 재정적 취약점(Bankruptcy Vector) 으로 기능하게 됩니다. 예를 들어, ‘결제 처리(Payment Processing)’ 권한을 가진 에이전트는 사기 탐지 임계값 아래에서 수천 건의 소액 결제를 반복 수행할 수 있으며, ‘읽기 전용(Read-Only)’ 권한만 가진 에이전트조차 고비용 API 엔드포인트를 지속적으로 호출해 서비스 거부(DoS) 공격을 발생시킬 수 있습니다.

에이전트의 행동을 이해하고, 자원 사용과 비용 한도를 자동으로 통제하며, 실행 맥락에 따라 신뢰를 지속적으로 평가할 수 있는 새로운 권한 부여 체계가 필요합니다.

1.2.2 결제 인프라의 불일치

기존 결제 시스템은 인간 중심의 거래 패턴을 전제로 설계되어 있으며, 이는 에이전트의 행동 양식과 정면으로 충돌합니다. 인간이 간헐적이고 규모가 큰 거래를 수행하는 반면, 에이전트는 지속적이고 연속적인 마이크로 결제 흐름(stream of micropayments) 을 생성합니다. 이러한 근본적 불일치는 에이전트의 자율성을 가로막는 경제적·운영적 장벽을 형성합니다.

에이전트 거래의 경제적 불가능성 **(Economic Impossibility of Agent Transactions)**

기존 결제 시스템의 단위 경제성(unit economics)은 에이전트 운영을 경제적으로 불가능하게 만듭니다. 신용카드 결제에는 건당 $0.30의 고정 수수료와 약 2.9%의 비율 수수료가 부과됩니다. 이러한 구조에서는 1센트(¢1)의 거래를 처리하는 데 30센트 이상의 비용이 발생하며, 이는 거래 금액보다 높은 수준입니다. 에이전트는 본질적으로 수많은 소액 거래를 생성하므로, 예를 들어 수천 건의 API 호출을 수행하는 에이전트는 실제 전송되는 가치보다 훨씬 큰 결제 수수료를 부담하게 됩니다.

이러한 경제적 제약은 에이전트의 효율성을 떨어뜨리는 비정상적 우회 방식(suboptimal workaround) 을 강요합니다. 대규모 선결제(Prepayment)는 자본을 묶어두고 정산(Reconciliation)을 복잡하게 만듭니다. 거래 묶음 처리(Batching)는 시간에 민감한 의사결정에서 위험한 지연(Latency)을 초래합니다. 지연된 결제(Delayed Settlement)는 에이전트가 확인을 기다리는 동안 기회를 잃게 만듭니다. 이러한 타협의 결과는 모두 에이전트가 필요로 하는 실시간·사용량 기반(pay-per-use) 모델에서 멀어지게 합니다.

또한 AI 에이전트의 글로벌한 활동 범위는 결제 시스템의 지역적 현실과 충돌합니다. 국제 송금은 금액에 관계없이 $15~$50의 비용이 들고, 처리에는 수일이 소요됩니다. 세 대륙의 공급망을 동시에 조율해야 하는 에이전트에게 이러한 결제 마찰은 인간의 운영이라면 마비를 초래할 수준의 비효율입니다. 그러나 바로 이러한 규모야말로 현대 에이전트가 작동해야 하는 현실입니다.

프로그래머블 머니의 부재 **(Absence of Programmable Money)**

기존 결제 시스템은 단순한 원자적 거래(Atomic Transfer) — 즉, “A에서 B로 X를 이동시키라”는 명령만을 수행합니다. 그러나 에이전트는 현 시스템이 표현할 수 없는 복합적 조건 로직을 갖춘 프로그래머블 머니**(Programmable Money)** 를 필요로 합니다.

- 스트리밍 결제**(Streaming Payments)**: 리소스 소비량에 따라 동적으로 조정되는 지속적 결제 흐름
- 조건부 송금**(Conditional Releases)**: 암호학적 완료 증명(Proof of Completion)이 확인될 때만 자금이 해제
- 자동 분배**(Automatic Splits)**: 실시간 기여도에 따라 비율적으로 계산된 자금 분배
- 조건부 에스크로**(Escrow with Triggers)**: 오라클 조건이 충족될 때 자금이 자동 해제
- 재귀 결제**(Recursive Payments)**: 상위 트랜잭션이 하위 트랜잭션을 자동 생성

현재의 시스템에서는 이러한 결제 패턴을 흉내 내기 위해 복잡한 외부 인프라 계층이 필요하며, 각 계층은 추가적인 복잡성, 지연, 그리고 연쇄적 실패 지점을 만들어냅니다. 단순한 프로그래머블 로직이어야 할 구조가 웹훅(Webhook), 데이터베이스, 정산 스크립트로 이루어진 취약한 탑(Fragile Tower) 으로 변질됩니다.

1.2.3 신뢰와 프로그래머빌리티의 결여 **(The Trust and Programmability Void)**

가장 근본적인 한계는 검증 가능하고 프로그래머블한 신뢰(Verifiable, Programmable Trust) 의 부재에 있습니다. 현재 시스템은 완전한 접근과 완전한 차단이라는 이분법적 신뢰 모델만 제공합니다. 이러한 공백은 에이전트가 사용자 정의 경계 내에서 증명 가능한 자율성을 유지하며 작동하는 것을 불가능하게 만듭니다.

검증 불가능한 제약 집행 **(Unverifiable Constraint Enforcement)**

사용자가 에이전트에게 권한을 위임하더라도, 해당 에이전트가 실제로 주어진 제약 조건을 준수하고 있는지 검증할 방법이 없습니다. 예를 들어, 투자 에이전트가 “일일 손실 한도 5%”를 지키고 있다고 주장하더라도 사용자는 너무 늦기 전까지 그 규칙이 위반되었는지 알 수 없습니다. 에이전트는 본질적으로 블랙박스로 작동하지만, 현재 시스템은 제약 조건이 올바르게 설정되었는지, 위반이 발생하지 않았는지, 혹은 보고된 행위가 실제 수행된 행위와 일치하는지를 암호학적으로 증명할 수 있는 수단을 제공하지 않습니다.

사용자는 포트폴리오가 청산되고, 계좌가 비워지고, 데이터가 파괴된 후에야 침해 사실을 인지합니다. 하지만 이 시점에는 이미 되돌릴 수 없으며, 에이전트가 실제로 무엇을 했는지, 언제 보호 장치가 실패했는지에 대한 검증 가능한 기록조차 존재하지 않습니다.

프로그래머블 거버넌스의 불가능성 **(Programmable Governance Impossibility)**

오늘날 에이전트의 거버넌스는 개별 플랫폼마다 조각나 있어, 사용자가 통합된 정책을 정의하거나 집행할 방법이 없습니다. 필요한 것은 모든 서비스 전반에 걸쳐 일관되게 적용되는 프로그래머블 거버넌스입니다. 플랫폼별로 흩어진 제어 대신, 사용자는 글로벌 차원의 제약을 코드로 설정할 수 있어야 합니다. 예를 들어 “모든 에이전트의 일일 총 지출을 $1,000 이하로 제한”하거나 “$100 이상의 거래는 서비스 종류와 관계없이 사용자 승인이 필요”와 같은 정책을 암호학적으로 정의할 수 있어야 합니다.

이러한 통합 거버넌스가 없다면, 에이전트는 플랫폼 간 공백을 이용해 제한을 우회할 수 있습니다. 반면 프로그래머블 거버넌스가 구현되면, 모든 트랜잭션은 사용자 정의 정책과 대조되어 검증되며, 에이전트는 시스템 전반에 걸친 증명 가능한 안전장치 내에서 작동하도록 보장됩니다.

책임 추적성과 대응 수단의 부재 **(Accountability and Recourse Vacuum)**

에이전트가 오작동하거나 악의적으로 행동했을 때, 현재의 시스템에서는 책임 소재를 암호학적으로 검증할 수 없습니다. 대신 관리자는 수동적인 사후 조사(Manual Forensic Process) 에 의존해야 합니다. 불변 감사 로그(Immutable Audit Trail)가 존재하지 않기 때문에, 에이전트가 실제로 어떤 행위를 했는지, 그 행위가 설정된 규칙을 위반했는지, 오류의 원인이 모델의 비정상적 판단(Hallucination), 플랫폼 결함, 통합 버그, 혹은 사용자의 승인 범위 초과 중 어느 쪽이었는지를 확인할 방법이 없습니다.

암호학적 증거가 없는 환경에서는 시스템의 복구가 코드가 아닌 법적 절차에 의존하게 됩니다. 사용자는 악성 에이전트를 즉시 차단할 수 있는 자동 제어 장치도, 무단 거래를 되돌릴 수 있는

수단도, 복구 절차를 자동화할 프로그래머블 복구 경로도 갖지 못합니다. 결국 자율형 에이전트의 약속은 책임 추적성의 부재 속에서 법적 리스크로 귀결됩니다.

1.3 스테이블코인: 에이전틱 인터넷의 네이티브 통화 (Stablecoin: The Native Currency for the Agentic Internet)

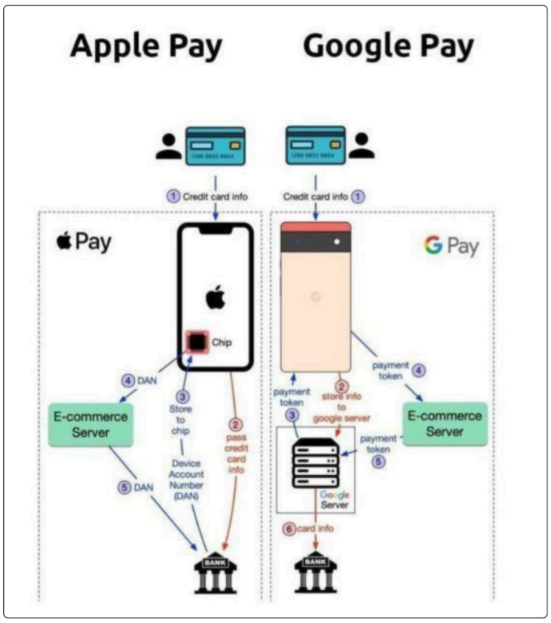
해결의 출발점은 '돈' 그 자체에 대한 근본적인 재정의에 있습니다. 지폐의 디지털 버전은 물리적 제약을 제거했지만, 에이전트가 직면한 경제적 비효율성과 지연 문제를 해결하지 못했습니다. 에이전트가 필요로 하는 것은 인간이 월별 명세서를 검토하기 위해 설계된 통화가 아니라, 기계가 검증할 수 있고, 프로그래머블하며, 요청 단위까지 측정 가능한 네이티브 통화입니다. 즉, 실시간으로 정산되는 스트리밍 마이크로 결제가 필요합니다.

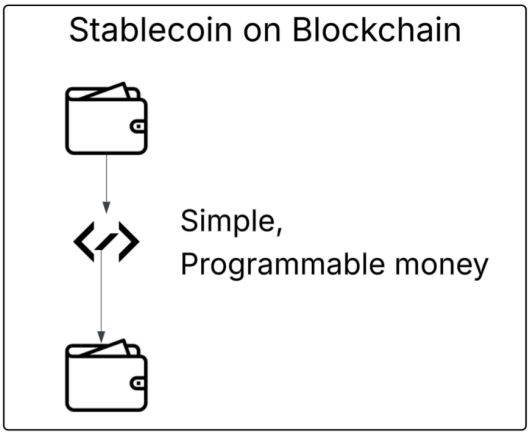
기존 결제 시스템은 에이전트 환경에서 완전히 실패합니다. 신용카드 결제 한 건은 발급사(Issuer), 결제 네트워크, 토큰화 서비스, 매입사, 은행 등 다수의 중개 단계를 거쳐야 합니다. 각 노드는 지연(Latency), 수수료(Fees), 실패 가능성(Failure Modes)을 추가합니다. 결과적으로 높은 고정 수수료, 수일에 걸친 정산 지연, 최대 120일까지 지속되는 차지백(Chargeback) 기간이 발생합니다. 이러한 구조에서는 에이전트가 요구하는 메시지 단위 결제(Pay-per-Message) 나 기계 간 자동 마이크로 결제(Machine-to-Machine Micropayment) 가 경제적으로 불가능합니다.

블록체인 기반 스테이블코인은 이러한 방정식을 근본적으로 뒤바꿉니다. 지갑 간 암호 서명된 스테이블코인 전송(Cryptographically Signed Stablecoin Transfer) 은 기존 시스템이 결코 달성할 수 없던 다음을 실현합니다.

- 거래 수수료 1센트 미만으로, 진정한 요청 단위 경제(Pay-per-Request Economics) 구현
- 차지백 리스크가 0인 즉시 확정성(Instant Finality)
- 조건부 결제와 스트리밍 결제를 위한 네이티브 프로그래머빌리티(Native Programmability)
- 국경 없는 상호운용성(Global Interoperability)과 결제 마찰 제거

이러한 이유로 스테이블코인은 단순한 결제 최적화 수단이 아닙니다. 기계 속도와 기계 정밀도로 작동하는 프로그래머블 가치 전송 수단으로서, 에이전틱 경제를 가능하게 하는 근본적 프리미티브(Fundamental Primitive) 입니다.





1.4 불완전한 혁명**:** 기존 결제 중심 블록체인으로는 충분하지 않다 **(The Incomplete Revolution: Why Existing Payments-First Blockchains Aren't Enough)**

최근 블록체인 아키텍처의 발전은 하나의 중요한 통찰에 도달했습니다. 결제와 일반 연산은 근본적으로 서로 다른 최적화를 필요로 한다는 것입니다. 이더리움은 프로그래머블 머니의 개척자였지만 그 설계는 결제 효율성보다는 최대 표현력**(Maximum Expressiveness)** 에 초점을 맞추었습니다. 그 결과, 가장 정교한 스마트 컨트랙트 플랫폼이 정작 단순한 결제조차 고비용·비예측적 수수료·혼잡한 블록 공간 문제로 어려움을 겪는 역설이 발생했습니다.

Paradigm이 제안한 **Tempo**는 블록체인 설계 철학의 중대한 전환점을 상징합니다. **Tempo**는 연산의 범용성보다 트랜잭션 효율성을 우선하는 결제 중심**(Payments-First)** 아키텍처입니다. 블록체인 활동의 약 **90%**가 복잡한 스마트 컨트랙트가 아닌 단순한 가치 이전**(Simple Value Transfer)** 에 해당한다는 현실을 인정하고, 이 전제를 기반으로 설계되었습니다. 그 결과, **Tempo**와 같은 결제 중심 체인은 비용·속도·예측 가능성 면에서 획기적인 개선을 달성했습니다.

Tempo의 아키텍처는 세 가지 핵심 최적화를 제공합니다.

- 예측 가능한 스테이블코인 수수료 **(Predictable Stablecoin Fees):** 스테이블코인으로 직접 수수료를 부과해 가스 토큰의 변동성을 제거
- 옵트인 프라이버시 **(Opt-in Privacy):** 거래는 기본적으로 비공개 상태를 유지하며, 필요 시 암호학적 증명으로 규제 준수 가능
- 전용 결제 레인 **(Dedicated Payment Lanes):** 결제 전용 블록 공간을 분리하여 다른 활동으로 인한 혼잡 방지

그러나 **Tempo**의 결제 중심 설계조차 자율형 에이전트 운영에 필요한 에이전트 네이티브 인프라**(Agent-Native Infrastructure)** 를 제공하지 못합니다**.**

- 지갑 계층 구조의 부재 **(No Wallet Hierarchy):** 에이전트와 사용자**(Principal)**의 계층이 분리되지 않아 보안 경계가 모호해집니다**.**
- 프로그래머블 안전장치 부재 **(Missing Programmable Guardrails):** 에이전트는 오류나 환각**(Hallucination)**을 일으킬 수 있으며**,** 이러한 손실을 방지할 수 있는 것은 암호학적 지출 규칙뿐입니다**.**
- 에이전트 트랜잭션 타입 부재 **(Absent Agent Transaction Types):** 결제 트랜잭션 내에 **API** 요청을 직접 내장**(Embed)**할 수 있는 네이티브 지원이 없습니다**.**
- 프로토콜 호환성 결여 **(Zero Protocol Compatibility): A2A, MCP, AP2, OAuth 2.1** 등 핵심 에이전트 표준과의 통합이 부족합니다**.**

이처럼 에이전트 특화 계층이 결여된 상태에서는 조직이 결제 권한을 에이전트에게 안전하게 위임 할 수 없습니다**.** 결과적으로 모든 에이전트 상호작용을 규제 준수하며 감사 가능한 마이크로 결제로 전환하겠다는 비전은 아직 실현되지 못했습니다**.**

Tempo는 결제 문제는 해결했지만**,** 에이전트 문제는 해결하지 못했습니다**.** 자율형 에이전트의 약속은 여전히 완전하게 실현되지 않았으며 에이전트 인프라 혁명은 아직 끝나지 않았습니다**.**

1.5 SPACE 프레임워크**:** 에이전트를 위한 엔드투엔드 인프라의 재정의 **(The SPACE Framework: End-to-End Infrastructure Reimagined for Agents)**

본 백서는 에이전틱 경제(Agentic Economy) 를 위해 제1 원리(First Principles) 에서 설계된 최초의 인프라 시스템, Kite를 소개합니다. 우리는 완전한 해법으로서 SPACE 프레임워크를 제시합니다.

- S – Stablecoin-native: 모든 트랜잭션은 스테이블코인으로 정산되며, 예측 가능한 1센트 미만의 수수료를 가집니다.
- P – Programmable constraints: 신뢰가 아닌 암호학적 강제(Cryptographic Enforcement) 를 통해 지출 규칙을 집행합니다.
- A – Agent-first authentication: 암호학적 주체 결합(Cryptographic Principal Binding) 을 갖춘 계층형 지갑 구조(Hierarchical Wallet Architecture) 를 채택합니다.
- C – Compliance-ready: 프라이버시 보호형 선택적 공개(Selective Disclosure) 와 불변 감사 로그(Immutable Audit Trails) 를 통해 규제 친화성을 확보합니다.
- E – Economically viable micropayments: 전 세계 규모로 요청 단위 실시간 결제(True Pay-per-Request Economics) 를 실현합니다.

Kite의 핵심 혁신

3계층 아이덴티티 아키텍처 **(Three-Layer Identity Architecture)**

Kite는 사용자(루트 권한), 에이전트(위임 권한), 세션(임시 권한)의 아이덴티티를 분리한 최초의 계층형 아이덴티티 모델을 도입합니다. 각 에이전트는 사용자의 지갑으로부터 BIP-32 방식으로 파생된 결정론적 주소를 부여받으며, 세션 키는 무작위로 생성되어 단 한 번만 사용 후 만료됩니다.

이러한 심층 방어 구조는 단계별 보안을 보장합니다. 세션이 침해되더라도 단 한 건의 트랜잭션만 영향을 받고, 에이전트가 침해되더라도 피해는 사용자가 설정한 제약 내로 제한됩니다. 사용자 키는 외부 접근이 불가능한 로컬 인클레이브에 안전하게 보관되며, 현실적으로 침해 가능성이 거의 없습니다.

자금은 보안을 위해 계층적으로 분리되지만, 평판은 시스템 전반에서 공유됩니다. 모든 트랜잭션과 상호작용은 통합된 평판 점수에 기여하며, 이는 사용자–에이전트–서비스를 아우르는 암호학적 신뢰 루트를 형성합니다.

스마트 컨트랙트를 넘어선 프로그래머블 거버넌스 (Programmable Governance Beyond Smart Contracts)

스마트 컨트랙트가 프로그래머블 머니를 가능하게 했다면, 에이전트에게 필요한 것은 여러 서비스 전반에 걸친 조합형 규칙(Compositional Rules) 입니다.

Kite는 사용자가 하나의 온체인 계정을 소유하고, 이 계정이 여러 검증된 에이전트에 의해 운영되는 통합형 스마트 컨트랙트 계정 모델(Unified Smart Contract Account Model) 을 구현합니다. 각 에이전트는 세션 키(Session Key)를 통해 암호학적으로 강제된 지출 규칙(Cryptographically Enforced Spending Rules) 하에서 작동합니다. 예를 들어: “ChatGPT 한도 월 $10,000, Cursor 한도 월 $2,000, 기타 에이전트 한도 월 $500.” 이러한 규칙은 시간 기반(Temporal), 조건 기반(Conditional), 계층 기반(Hierarchical)으로 정의될 수 있습니다. 예: “시간이 지남에 따라 한도를 점진적으로 증가시키거나, 시장 변동성이 커질 경우 한도를 절반으로 축소.” 이는 단순한 정책(Policy)이 아니라, 프로그램적으로 집행되는 경계(Programmatically Enforced Boundaries) 입니다.

에이전트 네이티브 결제 레일 **(Agent-Native Payment Rails with State Channels)**

스테이블코인이나 결제 중심 블록체인을 넘어, 혁신은 더 깊은 차원에서 시작됩니다. Kite는 에이전트 상호작용 패턴에 최적화된 프로그래머블 마이크로페이먼트 채널을 구축했습니다. 기존의 복잡한 결제 과정(인증, 요청, 결제, 대기, 검증)을 제거하고, 결제가 에이전트 상호작용 과정 내에서 즉시 처리되도록 설계되었습니다. 온체인 트랜잭션은 단 두 번(개시·종료)만 발생하며, 그 사이 수천 건의 오프체인 서명 업데이트가 처리됩니다. 이를 통해 100밀리초 미만의 지연과 100만 건당 1달러 수준의 비용으로 결제가 처리됩니다. 이러한 아키텍처적 전환은 요청 단위 및 스트리밍 마이크로결제를 핵심 동작으로 다루며, 센트 이하의 정밀도와 즉시 확정성을 구현합니다. 결과적으로, 이전까지 불가능했던 에이전트 네이티브 경제가 현실화됩니다.

다음 장에서는 Kite의 아키텍처, 프로토콜, 그리고 구현 방식이 앞서 언급된 인프라적 한계들을 어떻게 해결하는지를 자세히 다룹니다. 이로써 에이전틱 경제의 약속은 운영 가능한 현실로 전환됩니다.

에이전틱 미래는 더 나은 모델을 기다리는 것이 아닙니다.

에이전트를 진정한 경제 주체로 대우하는 인프라를 기다려왔습니다.

Kite와 함께, 그 기다림은 끝납니다.

2. Kite AI 설계 원칙

2.1 에이전틱 AI의 미래를 위한 맞춤형 설계 (Tailored Design for the Future of Agentic AI)

인간 중심 인프라에서 에이전트 중심 인프라로의 전환은 근본적인 아키텍처 재구성을 요구합니다. 비트코인에서 이더리움, 그리고 Tempo와 같은 결제 중심 블록체인에 이르기까지 모든 기존 블록체인은 “트랜잭션은 개인키를 관리하고 실시간으로 위험을 평가하며 필요 시 개입할 수 있는 인간이 시작한다”는 전제를 내포하고 있습니다. 이 전제는 인증 방식부터 정산 시점까지 시스템의 모든 설계를 규정합니다.

Kite는 이 전제를 근본적으로 뒤집습니다. Kite는 디지털 경제에서 AI 에이전트를 1급 시민(First-Class Citizen)으로 대우하기 위해 제1 원리에 따라 설계되었습니다. 각 에이전트는 고유한 암호학적 아이덴티티, 인증 체계, 지갑 계층 구조, 거버넌스 정책을 갖습니다. 이를 통해 에이전트는 인간의 개입이나 중간 과정을 거치지 않고, 네이티브하게 인증하고, 거래하며, 조율할 수 있습니다.

이 변화는 인프라의 모든 계층으로 확산됩니다. 수동 키 관리는 계층적 키 파생(Hierarchical Key Derivation) 으로 대체되고, 거래 검증은 프로그래머블 제약 집행(Programmable Constraint Enforcement) 으로 발전합니다. 사회적 평판은 암호학적 검증으로 대체됩니다. 이러한 설계 전환은 단순히 에이전트의 작동 방식을 바꾸는 것이 아니라, 인프라가 에이전트의 능력과 정렬될 때 무엇이 가능한가를 보여줍니다.

예를 들어 인간은 1,000달러의 거래를 승인하기 위해 금액을 읽고, 수신자를 확인하며, ‘확인’ 버튼을 누릅니다. 반면 에이전트는 동일한 거래를 처리하기 위해 자신이 허용된 권한 안에서 암호학적으로 인가되었음을 증명하고, 거래가 프로그래밍된 한도를 초과하지 않았음을 입증하며, 불변 감사 로그를 생성하고, 즉시 정산을 완료합니다. 인간의 승인 과정은 몇 초가 걸리지만, 에이전트의 과정은 밀리초 단위 연산으로 수행됩니다. 그러나 현재 인프라는 여전히 에이전트를 인간 중심 절차에 묶어두고 있어, 에이전트의 효율성과 자율성을 스스로 제한하고 있습니다.

2.2 암호학적 증명을 통한 신뢰 체인 (Chain of Trust with Cryptographic Proof)

Kite의 모든 행위는 암호학적으로 검증 가능한 감사 로그를 생성합니다. 이를 통해 사용자–에이전트–서비스–결과로 이어지는 불변의 신뢰 체인(Immutable Chain of Custody) 이 형성됩니다. 이 구조는 에이전트의 작동을 블랙박스 상태에서 벗어나게 하며, 모든 결정이 검증 가능하고 모든 제약이 수학적으로 집행되는 환경을 만듭니다.

Kite의 신뢰 체인은 세 가지 원칙으로 작동합니다.

직접 키 접근 금지**(No Direct Key Access)**

에이전트는 개인키에 직접 접근하지 않습니다. 각 작업은 단 한 번만 사용할 수 있는 작업 단위 세션 키(Task-Scoped Session Key) 를 부여받으며, 이 키는 필요한 권한만 정밀하게 갖습니다. 예를 들어 데이터 피드를 구매하도록 인가된 에이전트는 특정 공급자, 정확한 금액, 제한된 시간 내에서만 유효한 키를 사용합니다. 작업이 끝나거나 시간이 만료되면 이 키는 암호학적으로 무효화됩니다.

세분화된 작업 단위 권한 부여**(Fine-Grained Task Authorization)**

권한은 에이전트 전체가 아닌 작업 단위로 부여됩니다. 권한은 단순 지출 한도를 넘어 개별 API 엔드포인트, 데이터 유형, 조건 트리거 수준까지 제한됩니다. 이렇게 세밀하게 정의된 구조 덕분에 대규모 권한 탈취나 오용은 수학적으로 불가능합니다.

아이덴티티 노출 없는 평판 구조**(Reputation Without Identity Leakage)**

사용자와 에이전트는 독립된 아이덴티티를 유지하면서도 성과 기반 평판을 공유합니다. 각 사용자–에이전트 조합은 실행 이력에 따라 평판 점수를 축적하고, 이는 암호학적으로 사용자에게 귀속됩니다. 서비스는 에이전트 뒤에 실제 사용자가 존재함을 확인할 수 있지만, 사용자의 아이덴티티 정보는 노출되지 않습니다. 필요할 경우에만 선택적으로 소유권을 증명할 수 있으며, 기본값은 프라이버시가 보장된 상태입니다.

이 아키텍처는 투명성과 프라이버시를 동시에 보장하는 전례 없는 구조를 제공합니다. 규제 기관은 에이전트가 법적 한도 내에서 작동했음을 검증할 수 있고, 서비스는 결제 권한을 확인할 수 있으며, 사용자는 자신의 에이전트 활동 전체를 감사할 수 있습니다. 그러나 거래 로직과 개인 정보는 암호화된 상태로 보호됩니다. 시스템은 비밀을 공개하지 않고도 규제 준수를 증명합니다.

2.3 분리를 통한 주권 확보 (Sovereignty through Separation)

Kite는 탈중앙 자산 관리(Decentralized Asset Management) 와 개발자 서비스(Developer Services) 를 엄격히 분리해, 보안성과 사용성을 모두 확보하는 구조를 구현합니다.

탈중앙 자산 모델은 사용자의 완전한 주권을 보장합니다. 모든 자금은 스마트 컨트랙트 기반의 셀프 커스터디(Self-Custody) 지갑에 보관되며, Kite조차 이에 접근할 수 없습니다. 사용자는 자신의 디지털 자산을 독립적으로 통제하며, 에이전트는 수학적으로 정의된 제약 내에서만

작동합니다. 사용자는 Kite의 가용성과 무관하게 표준 블록체인 인터페이스를 통해 자산에 접근할 수 있습니다. 이는 약속이 아닌 암호학적 보증(Cryptographic Guarantee) 입니다.

Kite 플랫폼 서비스는 이러한 자산 관리 구조 위에서 작동하는 추상화 계층입니다. 플랫폼은 복잡한 암호 연산, 프로토콜 변환, 크로스체인 연동을 처리하며, 개발자는 익숙한 API 인터페이스를 통해 이를 활용합니다. 플랫폼은 키 파생, 세션 관리, 제약 설정을 자동화하지만, 자산에는 직접 접근하지 않습니다.

이러한 분리 구조는 블록체인 시스템의 근본적 긴장 관계—탈중앙화와 사용성 사이의 충돌—를 해소합니다. 완전한 탈중앙화는 종종 불편한 사용자 경험을, 완전한 중앙화는 단일 장애점(Single Point of Failure)을 초래합니다. Kite는 자산은 분산적으로, 서비스는 중앙화된 효율로 운영되도록 아키텍처를 분리함으로써 두 영역의 균형을 달성합니다.

이러한 구조의 함의는 기술적 아키텍처를 넘어섭니다. 규제 기관은 사용자 자금을 접근하지 않고도 플랫폼을 감사할 수 있으며, 개발자는 개인키를 직접 관리하지 않고도 정교한 애플리케이션을 구축할 수 있습니다. 사용자는 통제권을 포기하지 않고도 복잡한 연산을 위임할 수 있습니다. 각 이해관계자는 자신이 전문성을 가진 영역에서 독립적으로 작동하면서, 시스템은 모든 계층에서 일관된 보안성을 유지합니다.

2.4 네이티브 호환성**:** 기존 표준과의 제로 마이그레이션 마찰 **(Native Compatibility: Zero Migration Friction to Existing Standards)**

Kite는 생태계를 단절시키는 새로운 프로토콜을 만드는 대신, 기존 표준을 설계의 제**1** 원리로 받아들입니다. 이는 단순한 호환성 확보나 최소한의 규정 준수를 의미하지 않습니다. Kite는 다양한 표준을 아키텍처 수준에서 통합해, 여러 프로토콜을 동시에 네이티브하게 지원하는 시스템을 구축합니다.

Google의 **A2A** 프로토콜은 플랫폼 간 직접적인 에이전트 조율을 가능하게 합니다. Kite 에이전트는 A2A를 통해 이미 확립된 커뮤니케이션 패턴을 사용하여 다른 생태계의 에이전트와 자연스럽게 협력할 수 있습니다. 예를 들어, Kite 에이전트가 Google 에이전트와 함께 복잡한 워크플로우를 수행해야 할 경우, 중간 변환 계층이나 호환 브리지 없이 직접 통신이 가능합니다.

Anthropic의 **MCP**는 LLM 생태계 전체에서 모델 간 상호운용성을 보장합니다. Claude, GPT, 그리고 기타 신흥 모델들이 동일한 프로토콜 계층을 통해 상호작용함으로써, 모델 선택이 기술적 제약이 아닌 비즈니스적 의사결정으로 전환됩니다. 이를 통해 Kite 에이전트는 단일 워크플로우 내에서 Claude를 추론에, GPT를 생성에, 다른 특화 모델을 도메인별 작업에 활용할 수 있습니다.

OAuth 2.1 호환성은 인간 중심 서비스 및 기존 로그인·결제 흐름과의 자연스러운 연결을 의미합니다. 이를 통해 기존 서비스는 인증 시스템을 새로 구축할 필요 없이, 현재의 OAuth 구조에 Kite 에이전트를 쉽게 통합할 수 있습니다. 오늘 OAuth를 사용하는 서비스는 내일 단 몇 가지 설정 변경만으로 Kite 에이전트를 수용할 수 있습니다. 이러한 하위 호환성은 개발자에게

점진적인 마이그레이션 경로를 제공하며, 기존 인터넷과 엔터프라이즈 생태계 모두를 포용합니다.

X402 표준과 **Agent Payment Protocol**의 통합은 에이전트 네이티브 결제를 완벽하게 지원합니다. 특히 Kite 상에서는 스테이블코인 결제가 기본 결제 수단으로 사용되며, 향후 등장할 다양한 에이전트 결제 표준과의 호환성 역시 보장됩니다. 새로운 표준이 추가되더라도, Kite의 모듈형 아키텍처는 기존 기능을 손상시키지 않고 빠르게 통합할 수 있습니다.

이러한 표준 통합 접근법은 개발자와 서비스 모두에게 동일한 이점을 제공합니다. 개발자는 기존 스택을 유지하면서 Kite를 추가함으로써 시스템을 확장할 수 있고, 서비스 제공자는 새로운 인프라를 재구축할 필요 없이 기존 API에 에이전트 인식 기능만 추가하면 됩니다.

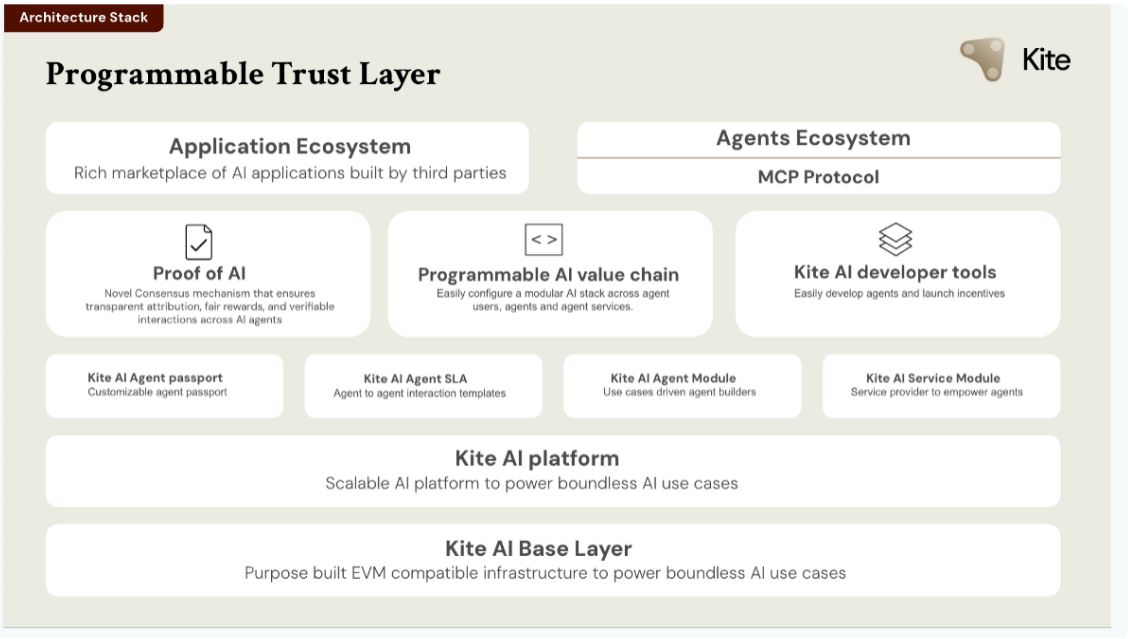
결국 미래의 인터넷은 단절이 아니라 진화(Evolution) 를 통해 완성됩니다. Kite는 그 진화를 자연스럽게 연결하는 네이티브 호환성의 토대(Native Compatibility Layer) 로 기능합니다.

3. 아키텍처

3.1 개요 (Overview)

Kite의 아키텍처는 전통적 블록체인 설계에서 벗어난 명확한 방향 전환을 보여줍니다. 모든 설계 결정은 자율형 에이전트가 수학적으로 보장된 안전성 아래에서 작동하도록 최적화되었습니다.

이 구조는 블록체인부터 애플리케이션까지 이어지는 목적 지향적 레이어로 구성되어 있으며, 각 레이어는 에이전트 자율성을 구성하는 핵심 과제를 해결합니다.



Base Layer: 스테이블코인 결제, 상태 채널, 결제 정산에 최적화된 EVM 호환 L1입니다. 범용 체인과 달리, 모든 최적화가 에이전트의 거래 패턴을 중심으로 설계되어 있습니다.

Platform Layer: 아이덴티티(Identity), 권한 부여(Authorization), 결제(Payments), SLA 집행(Enforcement) 을 위한 에이전트 준비형(Agent-ready) API를 통해 블록체인의 복잡성을 추상화합니다. 개발자는 익숙한 패턴으로 상호작용하고, 플랫폼은 암호학적 증명과 온체인 정산을 처리합니다.

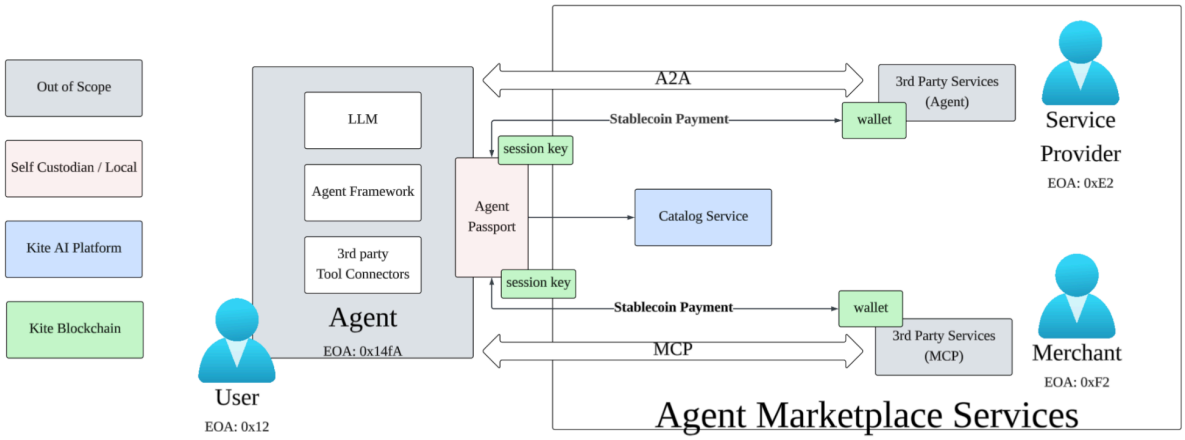
Programmable Trust Layer: Kite Passport(암호학적 에이전트 ID), Agent SLA(스마트 컨트랙트 상호작용 템플릿) 등의 신규 프리미티브와 A2A, MCP, OAuth 2.1에 대한 호환 브리지를 제공합니다. 이 레이어를 통해 에이전트는 Web3와 기존 서비스 모두와 매끄럽게 상호작용할 수 있습니다.

Ecosystem Layer: 표준 프로토콜로 연결된 에이전트 생태계와 AI 서비스를 위한 애플리케이션 마켓플레이스라는 두 개의 상호 연동 마켓플레이스로 구성됩니다. 서비스는 한 번 등록으로 수백만 에이전트에 의해 검색·활용될 수 있습니다.

이들 레이어는 함께, 에이전트가 예측 가능한 수수료로 거래하고, SLA를 강제하며, 계정과 상호작용 전반에 걸친 평판을 축적하게 만듭니다. Kite 아키텍처는 다음과 같은 변화를 실현합니다: 상태 채널을 통한 결제 비용의 급격한 절감, 전용 에이전트 레인을 통한 지연의 유의미한 개선, 암호학적 아이덴티티을 통한 자격증명 관리 오버헤드의 완전한 제거. 더 중요한 것은, 프로그래머블한 제약 준수 보장을 통해 진정한 에이전트 자율성이 구현된다는 점입니다.

3.2 용어와 핵심 개념 (Terminology and Core Concepts)

Kite를 이해하기 위해서는 자율형 에이전트 운영을 가능하게 하는 기본 구성 요소를 이해해야 합니다. 이들은 단순한 기술 정의가 아니라, AI 에이전트를 고도화된 챗봇에서 경제 주체(Economic Actor)로 전환시키는 아키텍처적 선택을 의미합니다.



3.2.1 핵심 주체

Kite 생태계는 네 가지 기본 주체를 중심으로 작동하며, 이들은 에이전트 경제의 핵심 역할을 담당합니다.

User (사용자**)**: 사용자는 다수의 AI 에이전트를 소유하고 제어하는 인간 주체**(Principal)** 입니다. 단순한 계정 보유자를 넘어, 전통적인 법적 체계와 자율 시스템을 연결하는 다리 역할을 수행합니다. 사용자는 에이전트에 특정 권한을 위임하되 최종 통제권을 유지하며, 암호학적 신뢰의 루트 역할을 하는 마스터 지갑을 관리하고, 모든 에이전트에 적용되는 글로벌 정책을 설정합니다. 법적으로는 여전히 사용자 본인이 책임 주체로 남기 때문에, 에이전트의 자율성은 결코 책임 부재를 의미하지 않습니다.

Agent (에이전트**)**: 에이전트는 사용자를 대신해 행동하는 개인형 **AI** 어시스턴트 또는 자율 프로그램입니다. 이들은 단순한 스크립트나 API 래퍼가 아니라, 복잡한 작업을 실행하고 다수의 서비스를 동시에 연동하며, 암호학적으로 제약된 경계 내에서 실제 자금을 처리할 수 있는 고도화된 존재입니다. 각 에이전트는 자체 지갑을 보유하고 독립적인 평판을 축적하며, 개별 거버넌스 정책 하에서 운영됩니다. 그러나 모든 에이전트는 BIP-32 파생 구조를 통해 사용자와 수학적으로 연결되어 있으며, 이를 통해 키 노출 없이도 명확한 소유 증명이 가능합니다.

Service (서비스**)**: 서비스는 에이전트가 작업을 수행하기 위해 상호작용하는 외부 제공 리소스입니다. 데이터 API, GPU 연산 제공자, SaaS 애플리케이션, 또는 다른 에이전트 기반 서비스 등 다양한 형태로 존재합니다. 서비스는 MCP(모델 인터페이스), A2A(에이전트 간

조율), OAuth(레거시 호환) 등 여러 프로토콜을 통해 통합됩니다. 각 서비스는 자체 접근 정책을 유지하면서도 글로벌 에이전트 마켓플레이스에 참여합니다.

Provider (서비스 제공자**)**: 서비스 제공자는 비즈니스, 개발자, 또는 인프라 운영자로서 수백만 개의 에이전트가 이용할 수 있는 서비스를 제공합니다. 이들은 단순히 API를 공개하는 데 그치지 않고, 자동 패널티가 적용되는 **SLA**를 정의하고, 검증 가능한 성과를 통해 평판을 구축하며, 모든 상호작용이 마이크로결제로 연결되는 프로그래머블 상거래 환경에 참여합니다. 결과적으로, 제공자(또는 판매자)는 기존의 복잡한 B2B 통합 과정을 플러그 앤 플레이형 에이전트 자원으로 전환시킵니다.

3.2.2 아이덴티티 및 신뢰 인프라

에이전트 경제에서의 아이덴티티는 단순한 사용자 이름과 비밀번호의 개념을 넘어섭니다. 에이전트가 작동하기 위해서는암호학적 증명, 검증 가능한 위임, 그리고 이동 가능한 평판(portable reputation) 이 필요합니다.

Kite Passport는 에이전트의 운영을 가능하게 하는 암호학적 아이덴티티 카드(cryptographic identity card) 입니다. 기존의 자격 증명이 단순히 인증을 수행하는 것에 그쳤다면, Passport는 사용자에서 에이전트, 그리고 행동(action)으로 이어지는 완전한 신뢰 사슬(trust chain) 을 형성합니다. 또한 Passport는 Gmail이나 Twitter와 같은 기존의 디지털 아이덴티티와 암호학적 증명을 통해 연결될 수 있어, 사용자가 자신의 기존 온라인 존재(digital presence)를 그대로 활용할 수 있게 합니다. Passport에는 단순한 아이덴티티 정보뿐 아니라 능력(capabilities)도 포함되어 있습니다. 즉, 에이전트가 무엇을 수행할 수 있는지, 얼마를 지출할 수 있는지, 어떤 서비스에 접근할 수 있는지가 정의됩니다. 가장 중요한 점은 선택적 공개(selective disclosure)가 가능하다는 것입니다. 에이전트는 자신이 검증된 인간 사용자에 속해 있음을 증명할 수 있지만, 그 사용자가 누구인지는 밝히지 않습니다. 이를 통해 프라이버시는 유지되면서도 책임성이 확보됩니다.

탈중앙 식별자(Decentralized Identifier, DID)는 에이전트와 사용자를 연결하는 전 세계적으로 고유하고 암호학적으로 검증 가능한 식별자입니다. DID는 임의의 문자열이 아니라 관계를 구조적으로 표현하는 계층적 식별자입니다. 예를 들어 사용자는 `did:kite:alice.eth` 를 가질 수 있고, 그녀의 트레이딩 에이전트는 `did:kite:alice.eth/chatgpt/portfolio-manager-v1` 과 같은 하위 식별자를 가질 수 있습니다. 이러한 계층 구조는 권한 체인을 즉시 검증 가능하게 만듭니다. 어떤 서비스라도 특정 세션이 실제로 한 에이전트에 속하는지, 그 에이전트가 특정 사용자로부터 위임받았는지, 그리고 해당 사용자가 현재 작업을 승인했는지를 암호학적으로 확인할 수 있습니다. 이 모든 것은 중앙 기관이나 API 호출 없이 수학적 증명만으로 이루어집니다.

검증 가능한 자격 증명(Verifiable Credentials, VCs)은 특정한 능력이나 권한을 입증하는 암호학적 증명서(cryptographic attestation) 입니다. VC는 에이전트가 컴플라이언스 교육을 이수했음을 증명하거나, 거래 라이선스를 보유하고 있음을 확인하거나, 평판 점수가 일정 기준 이상임을 증명할 수 있습니다. 이러한 VC는 자가 선언이 아니라 신뢰할 수 있는 발행 기관(trusted authority)이 암호 서명하여 발급합니다. VC는 세밀한 접근 제어(fine-grained access control)를 가능하게 하며, 서비스는 접근을 허용하기 전에 특정 증명을 요구할 수 있습니다. 예를 들어 거래소는 KYC 검증을, 데이터 프로바이더는 결제 이력을, 컴퓨팅

클러스터는 평판 점수를 요구할 수 있습니다. VC의 핵심은 조합 가능성(composability) 에 있습니다. 에이전트는 시간이 지남에 따라 다양한 자격 증명을 축적하며, 이를 통해 능력을 증명하는 이동 가능한 증거(portable proof of capabilities)를 형성합니다.

증명 체인(Proof Chains) / 감사 로그(Audit Trails)는 모든 에이전트의 행위가 블록체인에 불변(immutable) 하고 위·변조가 감지 가능한(tamper-evident) 형태로 기록되는 구조를 의미합니다. 이는 단순한 로그가 아니라, 실제로 어떤 일이 발생했는지를 보여주는 암호학적 증거(cryptographic proof of what occurred) 입니다. 이러한 체인은 사용자의 승인으로부터 에이전트의 의사결정, 그리고 최종 결과에 이르는 완전한 행위 계보(lineage) 를 확립합니다. 분쟁이 발생하면 증명 체인이 부인할 수 없는 증거를 제공하고, 규제 기관의 조사 시 완전한 투명성을 보장하며, 사용자는 자신의 에이전트를 감사할 때 모든 행동을 검증할 수 있습니다. 기존 시스템의 로그가 수정되거나 삭제될 수 있는 반면, Proof Chain은 과거 사건에 대한 수학적 확실성(mathematical certainty) 을 제공합니다.

3.2.3 지갑 및 결제 아키텍처

EOA 지갑(Externally Owned Account)은 개인키로 제어되는 전통적인 블록체인 지갑입니다. Kite 아키텍처에서 사용자는 권한의 루트 역할을 하는 마스터 EOA 지갑을 보유합니다. 이 지갑은 보안 인클레이브나 하드웨어 보안 모듈, 또는 사용자 기기 내에 안전하게 보관되며, 어떠한 상황에서도 에이전트나 외부 서비스에 노출되지 않습니다. EOA는 에이전트 운영에 필요한 특정 권한을 위임하기 위한 초기 권한 부여 서명을 수행해 신뢰 체인의 기초를 형성합니다. 개념적으로 단순하지만, EOA가 스마트 컨트랙트 위임과 결합될 때 그 기능은 강력하게 확장됩니다.

AA 지갑(Smart Contract Account)은 에이전트 결제를 가능하게 하는 혁신적 도약입니다. Account Abstraction(AA) 지갑은 단순한 주소가 아니라 내장 로직(built-in logic) 을 갖춘 프로그래머블 계정입니다. AA는 개발자가 스마트 컨트랙트를 통해 지출 규칙을 정의하고, 트랜잭션 실행을 번들링하며, 제3자가 가스비를 지불할 수 있게 합니다. 이러한 스마트 컨트랙트는 여러 계정과 상호작용하고, 프로그램 간 통신을 수행하며, 맞춤형 로직을 통합합니다. Kite의 모델에서 사용자는 온체인에 하나의 AA 계정을 보유하며, 이 계정에는 스테이블코인으로 표시된 공용 자금(shared funds)이 보관됩니다. 여러 에이전트가 세션 키를 통해 이 계정을 조작하지만, 각 에이전트는 인가된 한도 내에서만 작동합니다. 하나의 금고, 다수의 운영자, 완벽한 격리.

임베디드 지갑(Embedded Wallets)은 애플리케이션 내부에 직접 통합된 셀프 커스터디 지갑으로, 복잡성을 추상화하면서도 사용자 주권을 유지합니다. 사용자는 시드 구문이나 개인키를 직접 관리하지 않지만 자금에 대한 완전한 통제권을 유지합니다. 임베디드 지갑은 원클릭 에이전트 승인, 자동 세션 관리, 투명한 자금 흐름을 지원합니다. 이러한 지갑은 달러 단위로 사고하는 일반 사용자에게 블록체인을 보이지 않게 만들며, 동시에 에이전트 운영을 안전하게 보장하는 암호학적 보증을 유지합니다.

에이전트 결제 프로토콜(Agent Payment Protocol)은 기존 인프라에서는 구현할 수 없었던 결제 흐름을 가능하게 하는 포괄적 시스템입니다. 이 프로토콜은 센트 단위 이하의 마이크로결제, 사용량에 따라 지속적으로 이루어지는 스트리밍 결제, API 호출마다 가치가 부여되는 페이-퍼-인퍼런스(pay-per-inference) 모델, 그리고 성과 기반 조건부 결제를

지원합니다. 기존 신용카드는 최소 0.30달러의 고정 수수료가 부과되어 마이크로결제가 비경제적이며, ACH 송금은 정산에 수일이 소요되어 실시간 결제가 불가능합니다. 반면, 에이전트 결제 프로토콜은 기계 속도로 작동하는 즉시적이고 글로벌하며 프로그래머블한 가치 전송을 실현합니다.

온/오프 램프 API(On/Off-Ramp API)는 전통 금융과 에이전트 경제를 연결하는 가교 역할을 합니다. PayPal 및 주요 은행 파트너와의 통합을 통해, 사용자는 신용카드로 에이전트 지갑을 충전하고 상인은 수익을 은행 계좌로 인출할 수 있습니다. 이 램프는 컴플라이언스, 사기 방지, 통화 변환을 백엔드에서 자동으로 처리합니다. 사용자는 블록체인을 이해하지 않아도 됩니다. 그저 "달러가 들어오고 나가는" 사용자 경험만 제공합니다. 이러한 추상화 계층은 암호자산을 직접 보유하지 않는 수십억 명의 사용자에게도 에이전트 결제(Agentic Payments)를 손쉽게 이용할 수 있는 접근성을 제공합니다.

3.2.4 거버넌스 및 안전 메커니즘

자율형 에이전트는 기능적 역량을 유지하면서도 안전성을 보장할 수 있는 정교한 거버넌스 구조를 필요로 합니다.

SLA(Service-Level Agreement) 계약은 모호한 서비스 약속을 수학적으로 집행 가능한 보증으로 전환하는 스마트 컨트랙트입니다. 기존 SLA가 법적 절차에 의존했다면, 이 계약은 벌칙과 보상을 코드로 자동 실행합니다. 예를 들어 SLA는 99.9%의 가동 시간을 보장하며, 다운타임이 발생하면 자동으로 비례 환불을 수행합니다. 응답 시간이 100밀리초 이하일 경우 단계별 요금을 적용할 수도 있고, 데이터 정확도 기준을 설정해 오류 발생 시 자동으로 슬래싱이 이루어집니다. 이러한 프로그래머블 계약은 법정이 아닌 코드에 의해 신뢰를 형성합니다.

프로그래머블 트러스트 / 의도 기반 인가(Programmable Trust / Intent-Based Authorization)는 사용자가 자신의 의도를 수학적 제약으로 표현하고, 이를 블록체인에서 자동 집행되도록 컴파일하는 구조입니다. 이제 사용자는 에이전트가 정책을 따르길 '기대'할 필요가 없습니다. 시스템이 애초에 위반 자체를 불가능하게 만듭니다. 예를 들어, 사용자는 초과가 불가능한 지출 한도를 설정하거나, 특정 시간대 외의 작업을 자동으로 거부하는 시간 기반 제약, 특정 상점만 허용하거나 카테고리를 차단하는 화이트리스트·블랙리스트 규칙을 프로토콜 수준에서 강제할 수 있습니다. 또한 "변동성이 20%를 초과할 경우 한도를 절반으로 축소한다"와 같은 복합 조건 논리도 정의할 수 있습니다. 모든 의도는 일정 기간이 지나면 자동 만료되어, 잊힌 승인이 향후 취약점으로 남지 않습니다. 사용자의 의도는 더 이상 정책이 아니라 불변의 법칙으로 작동합니다.

세션 키(Session Keys) / 일시적 키(Ephemeral Keys)는 제로 트러스트 세션 관리를 구현하기 위한 임시 암호키입니다. 각 에이전트의 작업 단위마다 새로 생성되며, 영구 키로부터 파생되지 않기 때문에 완전 순방향 보안을 보장합니다. 예를 들어, 세션 키는 "오늘 오후 2시부터 2시 5분 사이에 공급자 A, B, C에게 데이터 피드 비용으로 최대 10달러를 송금"하도록 승인할 수 있습니다. 키는 인가된 작업이 완료되면 즉시 무효화되어 영구적으로 폐기됩니다. 설령 세션이 침해되더라도 피해는 몇 분 동안의 단일 작업, 제한된 금액으로 국한됩니다. 이러한 구조는 단 하나의 키 유출로 전체 시스템이 손상되는 기존 API 키 구조의 연쇄적 실패를 근본적으로 차단합니다.

평판 시스템(Reputation System)은 자기신고식 점수가 아닌 검증 가능한 행동 데이터를 기반으로 신뢰 점수를 축적합니다. 성공적인 결제는 평판을 높이고, 실패한 전달은 평판을 낮추며, 정책 위반은 즉시 패널티를 발생시킵니다. 그러나 이는 조작 가능한 주관적 평가가 아니라, 암호학적으로 증명된 실제 행동 데이터를 바탕으로 계산됩니다. 높은 평판을 가진 에이전트는 더 나은 요율, 높은 한도, 프리미엄 서비스 접근 권한을 부여받으며, 낮은 평판의 에이전트는 추가 검증이나 사용 제한을 받습니다. 평판은 서비스 간 이동이 가능하므로, 새로운 관계가 항상 '제로 트러스트'에서 시작되는 콜드스타트 문제를 해결합니다. 한 플랫폼에서 입증된 이력을 가진 에이전트는 다른 서비스에서 검증 가능한 자격 증명을 제시함으로써, 약속이 아닌 암호학적 증거에 기반한 신뢰 부트스트래핑을 실현합니다.

3.3 설계 구조

3.3.1 3계층 아이덴티티의 필연성 **(The Three-Layer Identity Imperative)**

전통적인 결제 시스템은 지불자(Payer)와 수취인(Payee)이라는 두 주체만을 인식합니다. 이러한 이분법적 구조는 사람이 직접 트랜잭션을 시작하고, 위험을 평가하며, 키를 관리하는 환경에서는 충분했습니다. 그러나 에이전트 시스템은 이 전제를 근본적으로 무너뜨립니다. 자율형 에이전트의 운영에는 키를 공유하지 않으면서도 권한을 위임할 수 있는 제3의 계층이 필요합니다.

사용자 아이덴티티 **(Root Authority)**

사용자는 암호학적 신뢰의 루트로서 최종적인 통제권을 유지합니다. 개인키는 보안 인클레이브, 하드웨어 보안 모듈(HSM), 또는 보호된 사용자 기기 내에 안전하게 저장되며, 에이전트나 서비스, 심지어 Kite 플랫폼조차 접근할 수 없습니다. 사용자는 단 한 번의 트랜잭션으로 모든 위임 권한을 즉시 철회할 수 있고, 모든 에이전트에 적용되는 글로벌 제약 조건을 설정하며, 불변의 증명 체인을 통해 모든 활동을 실시간으로 모니터링할 수 있습니다. 이는 서비스 약관이 보장하는 형식적 통제가 아니라, 암호학적 집행을 통해 보장되는 수학적 통제입니다.

에이전트 아이덴티티 **(Delegated Authority)**

각 AI 에이전트는 사용자의 지갑으로부터 BIP-32 계층 키 파생 구조를 통해 생성된 결정론적 주소를 부여받습니다. 예를 들어 사용자가 포트폴리오 관리를 위한 ChatGPT 기반 에이전트를 생성하면, 해당 에이전트는 사용자 지갑에 암호학적으로 연결되지만 독립적으로 분리된 주소를 가지게 됩니다. 이 주소는 에이전트의 온체인 아이덴티티로 작동하며, 결제나 자원 접근을 위한 세션 권한을 부여할 수 있습니다. 이 수학적 파생 구조는 강력한 특성을 제공합니다. 누구나 암호학적 증명을 통해 해당 에이전트가 특정 사용자에 속함을 검증할 수 있지만, 에이전트는 반대로 사용자의 개인키를 유추하거나 접근할 수 없습니다. 또한 에이전트는 자체 평판 점수를 유지하고, 다른 에이전트와 자율적으로 협업하며, 사용자가 정의한 제약 조건 내에서만 작동합니다. 심지어 에이전트가 완전히 침해되더라도 손실은 스마트 컨트랙트 제약 안으로 제한됩니다.

세션 아이덴티티 **(Ephemeral Authority)**

각 작업 실행 시 시스템은 완전히 무작위로 생성된 세션 키(Session Key)를 발급합니다. 이 키는 지갑이나 에이전트 키에서 파생되지 않기 때문에 완전한 순방향 보안(Perfect Forward Secrecy)이 보장됩니다. 세션 키는 단일 작업만 수행하는 일회성 권한 토큰으로, 영구 자격 증명을 노출하지 않습니다. 세션 키 생성은 전적으로 로컬 환경에서 수행되며, 서버 통신이나 개인키 전송은 발생하지 않습니다. 생성된 세션은 DID 세션으로 네트워크에 등록되며, 자체적으로 권한 증명, 신뢰 체인, 유효 기간을 포함합니다. 세션은 지정된 시간 윈도우 내에서만 유효하며, 만료 후에는 복원할 수 없습니다. 이는 양자 컴퓨터조차 복구할 수 없는 수준의 폐기성을 제공합니다.

이러한 3계층 아이덴티티 구조는 기존 결제 중심 블록체인이 갖지 못한 심층 방어(Defense in Depth)를 제공합니다. 세션이 침해되더라도 영향은 단일 트랜잭션으로 제한되고, 에이전트가 침해되더라도 사용자가 설정한 제약 범위를 넘을 수 없습니다. 오직 사용자 키가 침해된 경우에만 무제한 손실이 가능하지만, 보안 인클레이브 기반 보호 메커니즘으로 인해 그 가능성은 극히 낮습니다.

표준 기반 아이덴티티 해석 **(Identity Resolution through Standards)**

Kite는 ENS 표준을 확장하여 사람이 읽을 수 있는 식별자 형태로 에이전트 소유권을 증명할 수 있게 합니다.

- 사용자 ID: did:kite:alice.eth
- 에이전트 ID: did:kite:alice.eth/chatgpt/portfolio-manager-v1

퍼블릭 리졸버(Public Resolver)는 다음과 같은 즉각적 검증을 지원합니다.

- GetAgent(AgentID) → AgentID, AgentDomain, AgentAddress
- ResolveAgentByDomain(AgentDomain) → AgentID, AgentDomain, AgentAddress
- ResolveAgentByAddress(AgentAddress) → AgentID, AgentDomain, AgentAddress
- GetAgentBySession(SessionID) → AgentID, AgentDomain, AgentAddress, SessionInfo

서비스는 Kite나 사용자에게 직접 연락하지 않고도 완전한 권한 체인(Authority Chain)을 검증할 수 있습니다. 이를 통해 무허가형 상호운용성(Permissionless Interoperability)이 실현됩니다.

3.3.2 프로그래머블 거버넌스 및 화폐**(Programmable Governance and Money)**

스마트 컨트랙트가 '프로그래머블 머니'를 혁신했다면, 에이전트에게 필요한 것은 '프로그래머블 거버넌스'입니다. 이는 서비스 전반에 걸쳐 작동하며, 시간에 따라 진화하고, 외부 환경 변화에도 유연하게 대응할 수 있는 시스템입니다. 단순한 지출 한도를 넘어, 에이전트의 자율성을 안전하게 만드는 정교한 제어 구조입니다.

서비스 전반의 조합형 규칙 **(Compositional Rules Across Services)**

Kite의 제약 조건은 불리언 로직(Boolean Logic)을 기반으로 결합되어 복합 정책을 구성합니다. 예를 들어 "모든 플랫폼의 총 지출 < 일일 1,000달러 AND 단일 거래 < 100달러 AND 검증된

공급자만 허용"과 같은 규칙이 있습니다. 이러한 규칙은 스마트 컨트랙트 코드로 컴파일되어 원자적으로 평가되며, 에이전트는 트랜잭션을 분할하거나 서비스를 나눠서 한도를 우회할 수 없습니다. 블록체인은 최종 심판자로서, 수학적 확실성으로 규칙을 집행합니다.

신뢰의 시간적 진화 **(Temporal Evolution of Trust)**

고정된 한도는 시간이 지나며 관계가 형성되는 현실을 반영하지 못합니다. Kite는 '점진적 신뢰(Progressive Trust)' 모델을 도입합니다. 예를 들어 "초기에는 하루 10달러 한도로 시작하고, 신뢰가 쌓이면 매주 10달러씩 증가하여 최대 100달러/일 한도까지 상승"하는 방식입니다. 블록체인은 시간과 행동 데이터를 기반으로 자동으로 한도를 조정하며, 수동 개입이나 갱신 누락 없이 에이전트의 실제 성과를 반영하는 프로그래머블 신뢰 진화 구조를 제공합니다.

외부 신호에 대한 조건부 반응 **(Conditional Response to External Signals)**

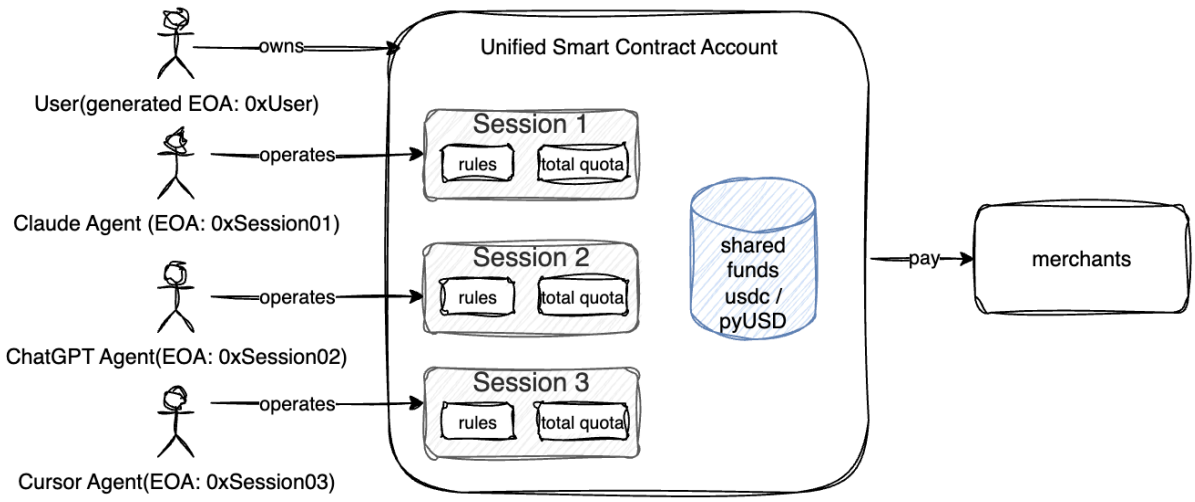
시장 상황은 변하고, 위협은 발생하며, 기회는 새롭게 등장합니다. 에이전트의 제약 조건 역시 이에 따라 동적으로 적응해야 합니다. 예를 들어 "변동성이 20%를 초과하면 거래 한도를 50% 축소한다"는 규칙을 설정할 수 있습니다. 오라클 네트워크는 실시간 시장 데이터를 스마트 컨트랙트로 전달하고, 자동 조정을 트리거합니다. 시장이 급변할 때 에이전트는 자동으로 보수적으로 전환되며, 보안 침해 발생 시 새로운 인가가 즉시 동결됩니다. 시스템은 인간보다 빠르게 위험에 대응합니다.

조직 단위의 계층적 거버넌스 **(Hierarchical Cascade Through Organizations)**

기업 환경에서는 위계적 거버넌스 구조가 필요합니다. 예를 들어 "ChatGPT 에이전트 월 한도 10,000달러, Cursor 2,000달러, 기타 에이전트 500달러"처럼, 하위 한도는 상위 계층의 제약 범위 내에서 자동 조정됩니다. 부서가 본부 예산을 초과할 수 없고, 개별 에이전트가 상위 티어의 할당량을 넘을 수 없는 구조입니다. 이러한 조직적 정책은 암호학적으로 전파되고 자동 집행됩니다.

통합 계정 모델 **(The Unified Account Model)**

Kite는 단순성과 통제력을 모두 충족하는 통합형 스마트 컨트랙트 계정 모델을 설계했습니다. 사용자는 USDC 또는 PYUSD로 공유 자금을 보유한 단일 온체인 계정을 소유하며, 에이전트들은 각각의 세션 키를 통해 이 계정을 운영합니다. 예를 들어 Claude, ChatGPT, Cursor와 같은 검증된 에이전트들은 각자의 세션(0xSession01~03)을 통해 할당된 규칙과 쿼터 내에서만 작업을 수행합니다. 에이전트가 작업을 실행하면 지출은 세션 허용 한도 내의 공유 풀에서 이루어지고, 계정은 판매자에게 프로그램적으로 결제를 수행합니다. 결과적으로 하나의 금고, 세션 단위의 리스크 분리, 그리고 모든 에이전트를 아우르는 정교하고 감사 가능한 통합 관리가 가능해집니다. 자금 분산도, 복잡한 정산 절차도 없습니다. 오직 하나의 구조적 계정 모델이 모든 것을 단순화합니다.



지출 규칙**(Spending Rules)** 및 정책**(Policy)**

시스템은 평가 방식(evaluation requirements)에 따라 온체인 규칙과 오프체인 정책을 구분합니다.

구분	지출 규칙	정책
적용 범위	자산 또는 스테이블코인 관련, 프로그래머블 머니(Programmable Money)를 활용	복잡한 로직 구현을 위한 완전한 제어력과 유연성 제공
평가 방식	스마트 컨트랙트를 통해 전적으로 온체인에서 평가되어 투명성 보장	사용자 로컬 환경 또는 Kite의 TEE(Trusted Execution Environment) 내에서 안전하게 오프체인 평가
상호 운용성	모든 온체인 계정과 통합 가능, 완전한 탈중앙화	플랫폼 종속적이나, 서드파티 시스템과 통합 가능

활용 사례	지출 한도, 롤링 윈도우(Rolling Window)	세션 TTL(Session Time-to-Live), 카테고리 제한, 수신자 목록 관리

Dimension	Spending Rules	Policy
Scope	Asset or stablecoin related, leveraging programmable money	Full control and flexibility for complex logic
Evaluation	Entirely on chain through smart contracts, ensuring transparency	Securely off chain in user's local or Kite's TEE
Interoperability	Integrates with any on chain account, fully decentralized	Platform specific but can integrate third party systems
Use Cases	Spending limits, rolling windows	Session TTL, categories, recipient lists

스마트 컨트랙트 기반 세션 키 구현

AA 지갑(Account Abstraction Wallet) 스마트 컨트랙트는 플러그인 아키텍처를 통해 세션 키를 구현합니다.

```
addSessionKeyRule(
    address sessionKeyAddress,  // The newly generated session key
    bytes32 agentId,           // Agent performing the actions
```

```
    bytes4 functionSelector,    // Allowed function (e.g., pay())
    uint256 valueLimit          // Maximum transaction value
)
```

사용자는 자신의 EOA(Externally Owned Account) 지갑으로 권한 부여 트랜잭션을 서명하며, 이는 암호학적 의도(cryptographic intent) 를 표현합니다. 이 트랜잭션은 UserOperation 형태로 패키징되어 번들러(bundler)를 통해 온체인에서 실행됩니다. 세션은 `validUntil` 타임스탬프가 만료되면 자동으로 종료되며, 사용자는 언제든 `removeSessionKeyPermission()`호출을 통해 직접 권한을 철회(revoke)할 수도 있습니다.

프로그래머블 에스크로 계약**(Programmable Escrow Contracts)**

Kite는 단순 결제를 넘어, 에이전트와 판매자 간 거래 사이에 스마트 컨트랙트를 배치하는 프로그래머블 에스크로 구조를 구현합니다. 구매자는 결제 의도에 서명하고, 자금은 만료 시간이 지정된 상태로 에스크로에 예치됩니다. 이후 결과에 따라 자금이 부분적으로 또는 전액으로 정산됩니다. 이 프로토콜화된 계약은 `authorize`, `capture`, `charge`, `void`, `reclaim`, `refund` 등 상거래 전 과정을 포괄하면서도, 비수탁형(non-custodial) 및 무허가형(permissionless) 구조를 유지합니다. 추가로 오퍼레이터 노드는 트랜잭션 제출 및 가스비 대납이 가능하지만, 항상 지불자의 서명된 해시에 의해 암호학적으로 제약됩니다. ERC-3009 표준은 서명 기반의 가스비 없는 사전 승인을 지원해, 완전한 사용자 주권 하에서 안전한 결제가 이루어집니다. 에스크로는 코드로 구현되므로, 매출 분배·토큰 스왑·프라이버시 모듈 등 다양한 비즈니스 로직 확장이 가능하며, 다른 스마트 컨트랙트와의 조합도 유연하게 지원합니다. 이로써 모든 트랜잭션은 프로그래머블하고, 감사 가능하며, 필요 시 복구 가능한 구조로 진화합니다

3.3.3 에이전트 프로토콜과의 호환성**(Compatible with Agent Protocols)**

현대의 에이전트는 고립된 상태로 작동하지 않습니다. 이들은 Google의 A2A(Agent-to-Agent) 프로토콜, Anthropic의 MCP(Model Context Protocol), 그리고 엔터프라이즈 환경의 OAuth 시스템과 통합되어 있습니다. 따라서 결제 인프라는 이러한 생태계와 자연스럽게 연결되는 네이티브 브리지(native bridge) 를 제공해야 하며, 임시적인 어댑터(adapter)에 의존해서는 안 됩니다.

프로토콜 변환**(Protocol Translation)**

에이전트 간 통신 표준 간의 매끄러운 매핑은 범용 상호운용성(universal interoperability)을 가능하게 합니다. 하나의 Kite 에이전트는 Google 에이전트와는 A2A로, Claude와는 MCP로, 엔터프라이즈 서비스와는 OAuth로 통신하며, 모두 동일한 아이덴티티(identity)와 결제 레일(payment rail)을 통해 작동합니다. 이는 미들웨어 수준의 번역이 아니라, 네이티브 다중 언어(multi-lingual) 능력을 가진 아키텍처입니다.

아이덴티티 연합**(Identity Federation)**

통합 인증(Unified Authentication)은 기존의 N×M 자격 증명 문제를 제거합니다. 사용자는 기존 제공자(provider)로 한 번만 인증하면 되고, 그 권한(permission)은 연결된 모든 서비스로 자동 상속됩니다. 중복된 자격 증명이나 동기화 문제는 사라지고, 단일하고 우아한 아이덴티티 흐름이 형성됩니다.

컴플라이언스 프레임워크**(Compliance Frameworks)**

전 세계의 다양한 규제 요건을 아우르는 내장형 컴플라이언스 프레임워크는 글로벌 운용성을 보장합니다. 유럽의 GDPR, 캘리포니아의 CCPA, 중국의 데이터 주거성(data residency) 요구사항 등이 구성 가능한 모듈을 통해 자동으로 처리됩니다.

통합은 다음과 같은 확장된 프로토콜 정의를 통해 이루어집니다.

```
// AgentDID definition
message AgentDID {
 string did = 1;                    // Unique identifier for the agent
 string user_wallet = 2;            // User's wallet address who controls this agent

 repeated VerificationMethod verification_methods = 3;
 AgentProvider provider = 4;

 int64 created = 5;                 // Timestamp of DID creation
 int64 updated = 6;                 // Last update timestamp
}

message VerificationMethod {
 string id = 1;                     // Verification key identifier
 string type = 2;                   // Always 'EcdsaSecp256k1VerificationKey2019'
 string controller = 3;             // DID that controls this key
 string public_key_hex = 4;         // Agent's public key for verification
}
```

A2A 프로토콜과 통합되었을 경우:

```
// AgentCard conveys key information: Key information uses a bold and larger fond.
message AgentCard {
// New DID we proposed.
 AgentDID id = 19;
 // The service provider of the agent. → like ChatGPT
 AgentProvider provider = 4;
 ......
 // description of agents
 // JSON Web Signatures computed for this AgentCard.
 repeated AgentCardSignature signatures = 17;
}

// AgentCardSignature represents a JWS signature of an AgentCard.
// This follows the JSON format of an RFC 7515 JSON Web Signature (JWS).
message AgentCardSignature {
 // The protected JWS header for the signature. This is always a
 // base64url-encoded JSON object. Required.
 string protected = 1 [(google.api.field_behavior) = REQUIRED];
 // The computed signature, base64url-encoded. Required.
 string signature = 2 [(google.api.field_behavior) = REQUIRED];
 // The unprotected JWS header values.
 google.protobuf.Struct header = 3;
}
```

이러한 아키텍처를 통해 Kite 에이전트는 각 주요 에이전트 프로토콜 내에서 1급 시민(first-class citizen) 으로 참여하면서 동시에 자신만의 고유한 기능과 확장성을 유지합니다.

3.3.4 프로그래머블 마이크로페이먼트 채널(Programmable Micropayment Channels)

직접적인 자금 전송은 간헐적 결제에는 적합하지만, 에이전트의 상호작용 패턴에서는 치명적으로 실패합니다. 에이전트는 메시지, API 요청, 추론 질의(inference query) 등 수천 건의 미세하고 빈번한 호출을 수행합니다. 기존 결제 방식에서는 전송된 가치보다 수수료가 더 커지는 역전 현상이 발생합니다. 상태 채널(State Channel)은 이러한 불가능한 경제 구조를 수익 가능한 현실로 전환합니다.

상태 채널은 두 참여자가 단 두 번의 온체인 트랜잭션만으로 오프체인에서 거래할 수 있게 합니다. 첫 번째 트랜잭션은 자금을 잠그기 위한 오픈(open), 두 번째는 정산(settlement)을 위한 클로즈(close)입니다. 이 두 앵커 사이에서 참여자들은 수천 건의 서명된 상태 업데이트를 즉시 교환합니다. 이를 통해 초당 수천 건의 트랜잭션 처리, 낮은 지연(latency), 수백만 건의 상호작용에 대한 수수료 분산(amortization)이 가능해집니다. 1밀리초 이하의 응답 시간을 요구하는 AI 추론(inference) 작업이 100만 건당 1달러 수준으로 경제성을 확보할 수 있는 이유가 바로 이 프로그래머블 마이크로페이먼트 채널입니다.

3.3.4.1 상호작용 패턴별 채널 유형(Channel Variants for Every Pattern)

Kite는 다양한 에이전트 상호작용 패턴에 최적화된 다수의 채널 타입을 구현합니다.

단방향 채널**(Unidirectional Channels)** 은 사용자에서 상인(merchant)으로 단일 방향으로 가치를 전송하며, 간단한 사용량 기반 과금(metering)에 적합합니다. API 소비, 데이터 피드, 추론 요청처럼 가치 흐름이 한 방향으로만 이동하는 경우 이상적입니다.

양방향 채널**(Bidirectional Channels)** 은 환불(refund), 크레딧(credit), 양방향 가치 교환(two-way value exchange)을 가능하게 합니다. 서비스가 에이전트에게 데이터를 제공하고 대가를 지급하거나, 오류 발생 시 자동 환불이 이루어지는 구조에 적합합니다.

프로그래머블 에스크로 채널**(Programmable Escrow Channels)** 은 상태 전이(state transition)에 맞춤형 로직을 내장합니다. EVM 개발자는 조건부 지급, 다자간 분배, 시간 잠금형 베스팅(time-locked vesting) 등의 임의 규칙을 정의할 수 있으며, 채널 자체가 미니 스마트 컨트랙트처럼 동작합니다.

가상 채널**(Virtual Channels)** 은 새로운 온체인 계약 없이 중간 노드를 통해 가치를 라우팅합니다. 예를 들어 에이전트 A가 허브 B를 통해 에이전트 C에게 결제함으로써, 추가적인 설정 없이 네트워크 효과를 실현할 수 있습니다.

프라이버시 보호 채널**(Privacy-Preserving Channels)** 은 상호작용의 기밀성을 보장합니다. 온체인에는 오픈(open)과 클로즈(close)만 기록되며, 수천 건의 마이크로 결제는 참여자 간 비공개로 유지되어 경쟁 정보와 사용 패턴을 보호합니다.

결과적으로 모든 메시지가 센트 단위 이하의 정밀도(sub-cent precision), 즉시 확정성(instant finality), 최소한의 온체인 자원 소모(on-chain footprint)로 결제됩니다. 이는 에이전트 기반 사용량 결제(pay-per-use) 및 스트리밍 경제(streaming economics)에 완벽히 부합합니다.

3.3.4.2 에이전트 경제에서 상태 채널의 한계가 강점이 되는 이유**(How State Channel Limitations Become Advantages in Agent Economy)**

상태 채널은 이더리움 기원에서 비롯된 여러 제약을 가지고 있습니다. 그러나 에이전트의 사용 패턴은 이러한 제약을 오히려 기능적 이점으로 전환합니다.

채널 오픈**/**클로즈 오버헤드**(Open/Close Channel Overhead)**

전통적 사용자는 채널을 자주 열고 닫지 않기 때문에 초기 비용이 비효율적입니다. 그러나 에이전트는 짧은 시간 동안 동일 서비스에 수백 건의 추론 요청을 보내므로, 설정 비용이 활동 밀도에 의해 완벽히 상쇄됩니다. 오버헤드는 사실상 무시 가능한 수준으로 전환됩니다.

생존 가정**(Liveness Assumption)**

채널은 참여자가 분쟁을 다투기 위해 온라인 상태를 유지할 것을 전제로 합니다. 인간 사용자는 예측 불가능하게 오프라인이 되지만, 평판이 걸린 전문 서비스(professional service)는 마이크로페이먼트를 위해 사기를 시도할 이유가 없습니다. 게임이론적으로 정직한 행동이 항상 우위에 있습니다.

그리핑 공격**(Griefing Attacks)**

악의적인 참여자가 상대에게 비용이 큰 분쟁 응답을 유도할 수 있습니다. 그러나 에이전트와 서비스는 평판 점수를 유지하며, 그리핑은 미미한 이익에 비해 평판을 파괴하기 때문에 경제적으로 비합리적입니다.

사전 정의된 참여자 집합**(Predefined Participant Sets)**

채널의 참여자는 개설 이후 변경할 수 없습니다. 그러나 에이전트 상호작용은 본질적으로 고정된 관계를 전제로 합니다. 사용자, 에이전트, 서비스 — 이 관계는 태스크가 시작되는 시점에서 이미 확정됩니다.

병렬 트랜잭션 처리**(Parallel Transaction Processing)**

상태 채널은 순차적(sequential) 업데이트를 처리하기 때문에 턴 기반(turn-based) 애플리케이션에 적합합니다. 에이전트 상호작용 또한 본질적으로 턴 기반으로, 요청(Request) → 응답(Response) → 재요청의 형태를 띕니다. 따라서 이 제약은 오히려 이상적으로 부합합니다.

3.3.5 전용 스테이블코인 결제 레인**(Dedicated Stablecoin Payment Lane)**

범용 블록체인은 결제를 수많은 트랜잭션 중 하나로만 취급하며, NFT 민팅, DeFi 스왑, 스마트 컨트랙트 배포와 동일한 블록 공간(block space)을 놓고 경쟁하게 만듭니다. 이는 경제적 모순을 초래합니다. 중요한 결제 트랜잭션이 투기적 거래 뒤에서 대기하며, JPEG의 인기에 따라 수수료가 요동칩니다. 단순한 $10 결제가 혼잡 시 $50의 가스비를 요구할 수 있고, 결제 시점과 정산 시점 사이에 가스 토큰의 가치가 변동되기도 합니다. 은행 스타일의 메타데이터는 취약한 오프체인 시스템에 덧붙여지며, 프라이버시는 항상 사후적 고려 사항으로 남습니다. 결과적으로 결제 시스템은 결제 시스템으로서의 기능을 상실합니다.

Kite는 이에 대한 근본적 혁신으로, 전용 결제 프리미티브(payment primitive) 와 자체적인 패스트 레인(fast lane) 멤풀 및 수수료 시장(fee market) 을 도입합니다. 이는 기존 시스템의 단순 최적화가 아니라, 결제 자체가 1급 인프라(first-class infrastructure)로서 취급되어야 함을 전제로 한 재설계입니다. 화이트리스트된 스테이블코인만이 송금 자산이자 수수료 지불 수단으로 사용되며, 실제 화폐 기준의 예측 가능한 비용을 보장합니다. 시스템은 결제(payment)와 단순 전송(transfer)을 명확히 구분하고, 결제 메모 필드에 은행급 메타데이터를 네이티브하게 지원하며, 온체인 상에서 컴플라이언스 정책을 강제하고, 배칭(batch) 및 상태 채널 마이크로페이먼트를 핵심 프리미티브로 통합합니다.

3.3.5.1 아키텍처 원칙

결제 레인은 블록체인 결제를 부수 기능에서 핵심 인프라로 탈바꿈시키는 다섯 가지 원칙을 기반으로 작동합니다.

1. 독립적 패스트 레인**(Independent Fast Lane)**: 별도의 멤풀과 수수료 시장이 존재해 결제 트랜잭션이 일반 거래와 블록 공간을 두고 경쟁하지 않습니다. 각 블록은 결제 전용 할당량을 보장합니다.
2. 스테이블코인 전용성**(Stablecoin Exclusivity)**: 거래 금액과 수수료는 화이트리스트에 등록된 스테이블코인으로만 지불됩니다. 이를 통해 결제 확인 과정에서 발생할 수 있는 토큰 가격 변동 문제를 제거합니다.
3. 결제 의미론**(Payment Semantics)**: 프로토콜은 커미션, 정산 데이터, 비즈니스 로직이 포함된 결제와 단순한 가치 이동(Transfer)을 명확히 구분합니다.
4. 은행 표준 정렬**(Banking Alignment)**: 송장 번호, 상점 분류 코드(MCC), 구매 주문 참조, 목적 코드 등 은행 결제에서 사용되는 필드를 네이티브하게 지원해, 블록체인 결제를 기존 금융 수준으로 끌어올립니다.
5. 에이전트 네이티브 기능**(Agent-Native Capabilities)**:
6. 결제 프리미티브 위에 인텐트 참조, 결제 위임, 장바구니, 마이크로페이먼트 채널을 유기적으로 결합해, 에이전트가 고도화된 결제 자동화를 수행할 수 있도록 합니다.

3.3.5.2 핵심 결제 연산 (Core Payment Operations)

결제 레인은 결제 전반의 요구를 포괄하는 다섯 가지 주요 연산을 구현합니다.

3.3.5.2.1 sendPayment

기본 결제 연산은 전용 결제 멤풀(Payment Mempool) 을 통해 라우팅되며, 풍부한 의미 체계(semantic support)를 제공합니다.

```
// Request Structure

Request
{
  "from": "0xPayer",
  "to": "0xMerchant",
  "asset": "0xUSDC",                // only whitelisted stablecoins
  "amount": "100.00",          // decimal string, parsed by asset decimals
  "feeAsset": "0xUSDC",             // pay fees in stablecoin
  "maxFee": "0.20",           // cap across base+tip to prevent overcharge
  "commission": {                   // payment-only
    "bps": 50,                      // 50 bp = 0.5%
    "recipient": "0xAggregator" // where commission goes (channel/platform)
  },
  "memo": {
    "text": "Invoice #128",
    "ref": "PO-2025-018",
```

```
    "mcc": "5732",          // Merchant Category Code
    "purpose_code": "GDSV"  // optional: aligns with bank/compliance codes
  },
  "policy": {
    "allowlist_id": "allow-v1",   // must match to enter chain
    "denylist_id": "sanctions-v3" // if matched, reject
  },
  "privacy": {
    "mode": "none|stealth|shielded",
    "view_key": "optional-view-key",
    "hint": "optional-addr-hint"
  },
  "deadline": 1768600000,         // unix seconds
  "nonce": "0x...",               // replay protection
  "intent_ref": "ap2:mandate:xyz",// bind to agent intent/mandate
  "signature": "EIP-712-signature"
}



// Response Structure

{
  "paymentId": "pay_0xabc...",
  "txHash": "0x...",
  "status": "accepted"            // accepted | rejected(policy/gas/asset)
| queued
}
```

3.3.5.2.2 sendTransfer

수수료가 없는 단순 송금은 결제 메모리풀을 사용하지만 구조적으로 간소화된 형태로 처리됩니다. 동일한 수수료 및 정책 규칙이 적용되며, 커미션이나 복잡한 정산 필드는 포함되지 않습니다.

3.3.5.2.3 estimatePaymentFee

기본 수수료, 우선순위 수수료 범위, 예상 커미션, 빠른 결제 레인 지연 시간에 대한 예상값을 반환합니다. 이를 통해 트랜잭션 제출 전 결제 비용을 사전에 예측할 수 있습니다.

3.3.5.2.2 sendTransfer

커미션이 없는 단순 전송(plain transfer)은 결제 멤풀을 공유하지만, 보다 단순한 의미 체계를 사용합니다. 동일한 수수료 및 정책 규칙이 적용되며, 커미션이나 복잡한 정산 필드는 포함되지 않습니다.

3.3.5.2.3 estimatePaymentFee

`base_fee`, `priority_fee_range`, `commission_estimate`, `expected_slo_ms` 등을 반환하여 패스트 레인 지연 시간(SLO)에 따른 비용 예측을 지원합니다. 이를 통해 제출 전 비용을 미리 산출할 수 있습니다.

3.3.5.2.4 sendPaymentBatch

배치 결제(batch payment)는 여러 전송을 단일 트랜잭션으로 압축합니다. 온체인에서는 수신자 정렬(recipient sorting) 및 금액 압축(amount compression)을 통해 효율적으로 인코딩되며, BLS 집계 서명(aggregate signature) 으로 가스비를 획기적으로 절감합니다.

주요 특징**:**

- 항목별 서브 메모(Sub-Memo) 또는 공통 배치 메모(Batch Memo) 지원
- 급여 지급(payroll)과 같은 사례를 위한 선택적 원자성(atomicity) 보장 (모두 성공 또는 전체 롤백)
- 대규모 배치는 자동 샤딩(sharding)되어 머클 증명(Merkle Proof)을 통해 병렬 검증 수행

3.3.5.2.5 Payment Channels

상태 채널(State Channel)은 결제 레인과 네이티브하게 통합되며 세 가지 연산을 지원합니다.

- **openPaymentChannel:** 자금 잠금(lock)을 통한 SLO 보장
- **commitPaymentChannel:** 오프체인 상태 업데이트
- **closePaymentChannel:** 결제 레인을 통한 최종 정산

이를 통해 “모든 메시지가 하나의 결제”로 작동하는 경제 모델을 실현합니다. 채널 내부 상호작용당 약 $0.000001의 비용이 발생하며, 수백만 건의 마이크로 결제에 걸쳐 수수료가 완전히 상쇄(amortized)됩니다.

3.3.5.3 노드 및 컨센서스 아키텍처

결제 레인은 성능 및 보안 보증을 유지하기 위해 특수화된 노드 인프라를 필요로 합니다.

3.3.5.3.1 결제 멤풀

결제 멤풀은 고도화된 우선순위 구조를 갖춘 독립적 큐(Independent Queue)로 작동합니다.

구조 및 할당**(Structure and Quotas)**

각 블록은 전체 공간의 일정 비율(X%)을 결제 트랜잭션에 전용으로 예약합니다. 이러한 보장된 할당량은 네트워크 혼잡 시에도 결제가 대기 상태로 밀려나지 않도록 합니다.

큐 관리**(Queue Management)**

네 가지 서브 큐(Sub-Queue)가 서로 다른 연산 유형을 처리합니다.

- 표준 결제(커미션 포함)를 위한 Payment Queue
- 단순 가치 이동을 위한 Transfer Queue
- 다중 수신자 연산을 위한 Batch Queue
- 상태 채널 정산을 위한 Channel Queue

우선순위 알고리즘**(Priority Algorithm)**

트랜잭션은 마감 기한**(deadline)**, 바이트당 수수료**(fee per byte)**, 도착 순서**(FIFO)** 순으로 정렬됩니다. 이를 통해 시간 민감형 결제가 안정적으로 처리됩니다.

입력 필터링**(Admission Filtering)**

사전 검증 단계(pre-mempool check)에서 유효하지 않은 트랜잭션은 즉시 거부됩니다.

- 화이트리스트 외 스테이블코인은 차단
- 블랙리스트(Denylist) 수신자는 거부
- 잘못된 서명은 폐기
- 만료된 마감 기한은 필터링
- 수수료 부족 트랜잭션은 거부

3.3.5.3.2 수수료 시장

결제 레인은 안정적이고 예측 가능한 비용 구조를 위해 독립적인 수수료 시장을 구현합니다.

독립 **EIP-1559** 구현**(Separate EIP-1559 Implementation)**: 결제 전용 기본 수수료(Base Fee)는 일반 트랜잭션과 별도로 조정됩니다. DeFi 영역의 혼잡이 결제 비용에 영향을 주지 않으며, `base_fee_payment` 변수는 결제 전용 혼잡도를 추적합니다.

스테이블코인 정산 옵션**(Stablecoin Settlement Options)**:

- 옵션 **1:** 검증자(Validator)가 스테이블코인을 직접 수령합니다. 최소한의 프로토콜 변경으로 가능한 가장 단순한 방식입니다.
- 옵션 **2:** 프로토콜에 내장된 자동 시장 조성기(AMM)가 스테이블코인을 스테이킹 자산으로 자동 교환합니다. 기존 검증자 경제 모델을 유지하면서 결제 안정성을 보장합니다.

커미션 처리**(Commission Processing)**: 프로토콜 거버넌스는 커미션 기준점(basis points)에 상한을 설정합니다. 커미션은 원금(principal)이 아닌 수수료 추가분(fee addon)에서 공제되어, 정산 과정이 간결하고 투명하게 유지됩니다.

3.3.5.3.3 컴플라이언스 인프라

컴플라이언스는 세 개의 레지스트리를 유지하는 PolicyRegistry 시스템 컨트랙트를 통해 작동합니다.

StablecoinWhitelist: 토큰 주소와 구성을 매핑합니다.

```
{
  "0xUSDC": {
    "decimals": 6,
    "freezeable": true,
    "issuer": "0xCircle"
  }
}
```

RecipientAllowlist / Denylist: 검증 가능한 증명(Verifiable Proof)을 지원하는 주소 집합입니다. 각 항목은 개별 주소이거나, 효율적인 대량 업데이트를 위한 머클 루트(Merkle Root)일 수 있습니다.

RuleSet Engine: 여러 규제 규칙을 조합합니다.

- IP 또는 신고된 위치 기반의 지리적 제한
- 허용된 운영 시간(Time Window)
- 금액 또는 빈도에 따른 거래 한도
- 상점 유형별 카테고리 제한

트래블 룰**(Travel Rule)** 컴플라이언스**:** 프로토콜은 FATF의 Travel Rule과 IVMS101 표준에 맞게 설계되어 있습니다. 온체인에는 참조 해시(reference hash)만 저장되며, 평문 PII(개인식별정보)는 오프체인에 저장됩니다. 모든 데이터는 검증 가능한 연결(assertion linking)을 통해 감사 시 추적이 가능합니다.

3.3.5.3.4 프라이버시 메커니즘

세 가지 프라이버시 모드는 서로 다른 사용 시나리오 요구사항을 충족합니다.

None 모드**:** 공공 결제를 위한 완전 투명 모드입니다. 모든 세부 정보가 온체인에 표시되며, 공익 펀딩(public goods funding)이나 투명한 운영에 적합합니다.

Stealth 모드**:** 수신자는 각 결제마다 1회용 주소를 생성합니다. 보기 키(View Key)를 통해 선택적 공개가 가능하며, 잔액과 트랜잭션 그래프는 비공개로 유지되지만 감사 가능성(auditability)은 유지됩니다.

Shielded 모드**:** 영지식 증명(Zero-Knowledge Proofs)을 통해 송금 금액과 참여자를 완전히 숨깁니다. 스테이블코인 쉴드 풀(Shielded Pool)이 개인 결제를 처리하며, 선택적 공개 메커니즘을 통해 규제 준수(Compliance)를 유지합니다.

메모 계층화**(Memo Tiering):** 2단계 메모 구조는 공개 데이터와 비공개 데이터를 구분합니다.

- 공개 계층(Public Tier): 상인에게 보이는 정산용 데이터
- 비공개 계층(Private Tier): 보기 키 보유자만 열람 가능한 암호화된 데이터

3.3.5.4 Semantic Payment Types 의미적 결제 유형

프로토콜은 정확한 경제 지표 산출을 위해 결제 유형을 명확히 구분합니다.

Payments에는 다음이 포함됩니다.

- 플랫폼 및 애그리게이터를 위한 커미션 분배(Commission Split)
- 회계 시스템용 정산 메타데이터(Reconciliation Metadata)
- 에이전트 운영을 위한 인텐트(Intent) 및 위임(Mandate) 참조
- 전자상거래 결제 시 주문(Order) 및 장바구니(Cart) 바인딩

Transfers는 다음을 제공합니다.

- 단순한 P2P 가치 이동
- 커미션 오버헤드 없음
- 최소한의 메타데이터 요건
- 최대 처리량(Throughput) 최적화

노드, 익스플로러, 인덱서는 이러한 구분을 인식하여 총 상품 거래액**(GMV)**, **플랫폼 테이크 레이트(Take Rate)**등의 정확한 지표를 계산합니다.

3.3.5.5 최적화 전략

3.3.5.5.1 배치 처리

대규모 운영은 고도화된 배치(batch) 메커니즘을 통해 효율을 극대화합니다.

중복 제거 및 정렬**(Deduplication and Sorting):** 중복된 수신자를 제거하고, 최적 압축을 위해 정렬합니다. 금액은 이전 값으로부터의 델타 인코딩(delta encoding)을 통해 압축됩니다.

샤딩 전략**(Sharding Strategy):** 대형 배치는 머클 증명(Merkle Proof)으로 연결된 샤드로 분할되며, 노드는 병렬로 샤드를 검증해 처리량을 유지합니다.

실패 처리 옵션**(Failure Handling Options):**

- 부분 허용**(Partial Tolerance):** 실패한 수신자를 건너뛰고 성공한 항목만 처리
- 원자 모드**(Atomic Mode):** 모든 항목이 성공하거나 전부 되돌림 (Critical Operation에 사용)

내보내기 형식**(Export Formats):** 인덱서는 CSV 및 Parquet 파일을 생성하여 원클릭 정산(Reconciliation)을 지원하며, 표준 회계 소프트웨어에서 바로 가져올 수 있습니다.

3.3.5.5.2 개발자 통합

EIP-712 Signing Standard:

```
const domain = {
  name: "KitePayment",
  version: "1",
  chainId: 1,
  verifyingContract: "0xPaymentLane"
}
```

Payment, Transfer, BatchItem 각각에 대한 별도 타입 정의를 통해 교차 도메인 재생 공격을 방지합니다.

SDK (SDK Methods):

- `signPayment()`: 결제 서명 생성
- `signBatch()`: 배치 결제 서명 생성

- `simulatePayment()`: 제출 없이 실행 테스트
- `watchPaymentReceipt()`: 결제 확정 상태 모니터링
- `exportReconciliation()`: 회계 보고서 생성

3.3.5.6 에이전트 네이티브 기능

결제 레인은 고도화된 에이전트 행위를 가능하게 하는 핵심 프리미티브를 제공합니다.

인텐트 및 위임 시스템**(Intent and Mandate System):** 모든 결제는 자신을 승인한 인텐트(Intent)를 참조합니다. 스마트 컨트랙트는 결제가 승인된 한도, 카테고리, 시간 조건 내에서 실행되었는지를 검증합니다. 모든 제한과 규칙은 자동으로 강제됩니다.

장바구니 통합**(Cart Integration):** 결제는 메모 필드를 통해 장바구니 스냅샷을 참조하며, 영수증에는 콘텐츠 지문(Content Fingerprint)이 포함됩니다. 분쟁 발생 시 결제 시점의 장바구니 내용을 암호학적으로 증명할 수 있습니다.

마이크로페이먼트 스트리밍**(Micropayment Streaming):** 에이전트 간 상호작용은 각 메시지에 결제 바우처(Payment Voucher)를 포함합니다. 정산은 N개의 메시지마다 또는 일정 시간 간격(T)에 자동으로 트리거되며, 채널 실패 시 온체인 정산으로 자동 전환됩니다.

프로그래머블 정산**(Programmable Settlement):** 템플릿을 통해 복잡한 자금 분배 로직을 정의할 수 있습니다.

- 플랫폼 수수료가 자동 공제
- 상인 몫은 비율적으로 분배
- 크리에이터 로열티는 재귀적으로 처리
- 세금 원천징수(Tax Withholding)는 적절히 예약

3.3.5.7 보안 및 리스크 관리

결제 레인은 공격 및 운영 리스크로부터 시스템을 보호하기 위한 종합적인 보안 메커니즘을 구현합니다.

서비스 거부**(DoS)** 방지 **(Denial of Service Prevention)**: 사전 검증(pre-admission validation) 단계에서 유효하지 않은 트랜잭션을 엣지 레벨에서 차단합니다. 과도한 부하를 유발하는 주소는 속도 제한(rate limiting)을 통해 제어되며, 스테이블코인 수수료 요건으로 인해 경제적 스팸은 사실상 불가능합니다.

MEV 저항성**(MEV Resistance)**: 우선순위 계층 내에서는 선착순 (First-Come First-Served) 순서가 적용되고, 마감 기한(deadline) 강제가 트랜잭션 홀딩을 방지합니다. 민감한 작업을 위한 프라이빗 멤풀(private mempool) 옵션도 제공됩니다.

스테이블코인 동결 처리**(Stablecoin Freeze Handling)**: 프로토콜은 발행자 동결 이벤트를 자동 탐지하며, 영향받은 자금을 '분배 불가 잔액(undistributable balance)'으로 표시합니다. 영수증에는 동결 상태가 기록되고, 장애가 발생해도 정산 정확성은 유지됩니다.

비상 통제**(Emergency Controls)**: 거버넌스는 특정 연산을 일시 중단할 수 있으며, 이상 거래량이 감지되면 서킷 브레이커(circuit breaker)가 자동 트리거됩니다. 치명적 장애 발생 시 롤백 메커니즘이 작동하여 시스템 안정을 보장합니다.

이러한 전용 결제 인프라는 블록체인 결제를 고비용의 부수 기능이 아닌, 효율적이고 규제 준수하며 프라이버시를 보장하는 핵심 운영 체계로 변모시킵니다. 그 결과 기존 금융 시스템과 동등하거나 그 이상을 달성하면서, 레거시 금융에서는 불가능했던 새로운 기능들을 가능하게 합니다.

3.4 프로그래머블 트러스트 기반 보안**(Security with Programmable Trust)**

자율형 에이전트의 신뢰는 암호학적으로 보장된 보안 체계 없이는 유지될 수 없습니다. 에이전트 침해가 무제한 손실로 이어진다면 사용자는 실질적인 권한을 위임할 수 없고, 검증 가능한 권한 부여가 없다면 서비스는 에이전트의 요청을 수락할 수 없습니다. 또한 감사 가능한 컴플라이언스가 없다면 규제 기관은 해당 운영을 승인할 수 없습니다. Kite의 보안 아키텍처는 기존 시스템이 제공하는 약속이 아닌, 수학적으로 증명된 확실성을 제공합니다.

3.4.1 핵심 암호 구성요소**(Core Cryptographic Components)**

Kite 프로토콜은 세 가지 상호 연동된 암호학적 구성 요소를 통해 사용자 의도에서 에이전트 실행까지 이어지는 완전한 권한 부여 체인을 형성하며, 침해가 불가능한 수준의 보안을 달성합니

3.4.1.1 Standing Intent: The Root of Authority Standing Intent: 권한의 루트**(The Root of Authority)**

Standing Intent(SI)는 에이전트가 수행할 수 있는 행위를 사용자가 암호학적으로 서명한 선언을 의미합니다. 이는 정책 문서나 설정 파일이 아니라, 위조나 초과가 불가능한 수학적 인가 증명입니다.

```
SI = sign_user(
    iss: user_address,      // Issuer: the authorizing user
    sub: agent_did,         // Subject: the authorized agent
    caps: {                 // Capabilities: hard limits
        max_tx: 100,        // Maximum per transaction
        max_daily: 1000     // Maximum daily aggregate
    },
    exp: timestamp          // Expiration: automatic revocation
)
```

사용자의 개인키로 서명된 Standing Intent는 변경 불가능한 신뢰의 루트(immutable root of trust)가 됩니다. 이후의 모든 연산은 유효한 SI로부터 추적되어야 하며, 설정된 capabilities는 에이전트 오류나 서비스 침해 상황에서도 초과될 수 없습니다. 만료 시점이 도래하면 인가가 자동 폐기되어 잊힌 권한이 영구적으로 남지 않습니다.

3.4.1.2 Delegation Token: 에이전트 권한 부여(Agent Authorization)

Delegation Token(DT) 은 에이전트가 영구 자격 증명 정보를 노출하지 않고, 특정 작업을 수행하기 위한 세션에 권한을 부여할 수 있도록 합니다.

```
DT = sign_agent(
    iss: agent_did,                // Issuer: the delegating agent
    sub: session_pubkey,           // Subject: the session key
    intent_hash: H(SI),            // Link: hash of Standing Intent
    operation: op_details,         // Scope: specific operation
    exp: now + 60s                 // Expiration: short lived
)
```

Delegation Token은 에이전트가 Standing Intent의 범위 내에서 해당 세션과 작업에 권한을 부여했음을 암호학적으로 증명합니다. 해시 연결 구조를 통해 에이전트는 사용자 한도를 초과할 수 없으며, 짧은 만료 시간은 세션이 침해되더라도 노출 범위를 최소화합니다. 또한 작업 범위를 제한함으로써 세션의 재사용을 원천적으로 차단합니다.

3.4.1.3 Session Signature: Execution Proof Session Signature: 실행 증명(Execution Proof)

Session Signature(SS)는 트랜잭션 실행의 최종 암호학적 증거를 제공합니다.

```
SS = sign_session(
    tx_details,  // Complete transaction data
    nonce,       // Replay prevention
    challenge    // Freshness proof
)
```

서비스는 모든 작업을 수락하기 전에 세 가지 서명을 검증합니다. Standing Intent는 사용자 인가를, Delegation Token은 에이전트 위임을, Session Signature는 실제 실행을 증명합니다. 이 3중 검증은 '금지된 행위'를 넘어 '암호학적으로 불가능한 행위'를 만듭니다.

3.4.2 증명 가능한 보안 속성**(Provable Security Properties)**

이 프로토콜은 적대적 환경에서도 시스템 동작에 대한 수학적 보증을 제공하여, 신뢰 기반 보안에서 증명 기반 보안으로 패러다임을 전환합니다.

정리 **1:** 손실 상한 **(Bounded Loss)**

명제**:** 권한 설정 C와 기간 D를 가진 Standing Intent SI가 주어졌을 때, 완전한 에이전트 침해 상황에서의 최대 추출 가능 가치 (MEV)는 다음과 같습니다.

MEV ≤ C.max_daily × D

증명**:** 프로토콜 구조상 모든 트랜잭션은 유효한 Standing Intent 서명을 요구합니다. 암호학적 서명 검증은 유효한 SI를 가진 트랜잭션만 실행되도록 보장합니다. 프로토콜이 강제하는 capabilities에 따라 단일 거래는 C.max_tx를, 일별 합계는 C.max_daily를 초과할 수 없습니다. 또한 Standing Intent는 기간 D 후 만료되므로 이후에는 트랜잭션이 실행될 수 없습니다. 따라서 최대 추출 가치는 일일 한도와 기간의 곱으로 한정됩니다.

이 정리는 사용자에게 정량적 리스크 지표를 제공합니다. 예를 들어 하루 $100 한도로 30일간 에이전트를 승인한 사용자는 최대 노출액이 정확히 $3,000임을 확신할 수 있습니다. 이러한 수학적 확실성은 실질적 금융 권한의 안심 위임을 가능하게 합니다.

정리 **2:** 위조 불가능성 **(Unforgeability)**

명제**:** 사용자의 개인키에 접근하지 못한 공격자는 무단 에이전트를 위한 유효한 Standing Intent를 생성할 수 없다.

증명**:** Standing Intent의 유효성은 ECDSA 또는 EdDSA 알고리즘을 사용하여 사용자의 공개키로 검증됩니다. 이 서명 기법이 선택된 메시지 공격에 대한 존재적 위조 불가능성(EUF-CMA)을 만족한다는 가정하에, 공격자는 개인키 없이 유효한 서명을 만들 확률이 무시할 수 있을 정도로 낮습니다. 이산 로그 문제의 수학적 난이도는 공개 정보로부터 개인키 복원을 계산적으로 불가능하게 합니다.

이 정리는 공격자가 위조 인가를 생성할 수 없음을 보장합니다. 기존 SI를 모두 알고 있고 트랜잭션을 관찰하며 침해된 에이전트를 통제하더라도, 새로운 무단 인가를 만드는 것은 불가능합니다. 사용자의 개인키만이 권한 위임의 유일한 근원입니다.

추가 보안 속성**(Additional Security Properties):** 세션 키 침해는 해당 세션의 작업에만 영향을 미치며, 과거와 미래 세션은 독립 키 생성으로 보호됩니다(전방 보안, Forward Secrecy). 권한 위임의 각 단계는 권한을 감소시켜, 세션은 에이전트보다 작은 권한을, 에이전트는 사용자보다 작은 권한을 갖습니다(최소 권한 원칙, Principle of Least Privilege). 모든 인가에는 만료 타임스탬프가 포함되어 잊힌 위임이 무기한 존속할 수 없습니다(자동 만료, Automatic Expiration). 마지막으로 모든 서명은 부인 방지(Non-Repudiation)를 보장하여, 사용자는 에이전트 인가를, 에이전트는 세션 인가를, 세션은 트랜잭션 실행을 부인할 수 없습니다.

3.4.3 철회 메커니즘(Revocation Mechanism)

권한 철회가 없는 보안은 완전할 수 없습니다. Kite 프로토콜은 침해되었거나 비정상적으로 작동하는 에이전트가 운영을 지속할 수 없도록, 다층적 집행 구조를 갖춘 포괄적인 철회 시스템을 구현했습니다.

3.4.3.1 즉각적 로컬 철회(Immediate Local Revocation)

사용자는 Kite Hub로 철회 메시지를 브로드캐스트하며, 이 메시지는 즉시 모든 등록된 서비스 제공자(provider)에게 전파됩니다. 이 P2P(피어 투 피어) 전파 구조는 블록체인 확인을 기다리지 않고도 거의 즉각적인 철회를 가능하게 합니다. 철회 알림을 수신한 서비스는 해당 에이전트의 모든 요청을 즉시 거부합니다.

철회 메시지에는 다음 정보가 포함됩니다.

- 철회 대상 에이전트 식별자(Agent Identifier)
- 철회 타임스탬프(Revocation Timestamp)
- 권한을 증명하는 사용자 서명(User Signature)
- 선택적 사유 코드(Optional Reason Code, 분석용)

네트워크 효과는 철회 속도를 가속화합니다. 연결된 서비스가 많을수록 전파 속도가 빠르고, 주요 서비스는 철회 메시지를 우선적으로 처리합니다. 대부분의 경우 네트워크 전체가 수 초 내에 철회를 인지합니다.

3.4.3.2 Cryptographic Revocation 암호학적 철회

사용자는 서비스 제공자가 캐싱하고 검증할 수 있는 철회 인증서(Revocation Certificate) 를 서명합니다.

```
RC = sign_user(
    action: "revoke",
    agent: agent_did,
    standing_intent: SI_hash,
    timestamp: now,
    permanent: true/false
)
```

네트워크 전파가 실패하더라도, 이 암호학적 철회 증명은 에이전트의 지속 작동을 방지합니다. 서비스는 요청을 수락하기 전에 항상 철회 인증서를 확인하며, 캐싱된 인증서는 재시작 이후에도 유지됩니다. 영구 철회로 표시된 경우 복구는 불가능하며, 이는 강력한 보안 보증을 제공합니다.

3.4.3.3 Economic Revocation 경제적 철회

에이전트 채권(agent bond)에 내장된 슬래싱(Slashing) 조건은 철회 이후 작동을 지속하려는 행위에 대해 강력한 경제적 억제력을 제공합니다.

에이전트 채권**(Agent Bonds):** 에이전트는 정상적 행위를 보증하기 위해 일정량의 토큰을 담보로 예치합니다. 채권 규모는 인가 한도(authorization limit)에 비례하여 설정되어, 올바른 행동에 대한 인센티브 정렬이 이루어집니다.

슬래싱 트리거**(Slashing Triggers):** 철회 이후 작동이 감지되면 프로토콜이 자동으로 슬래싱을 실행합니다. 프로토콜은 철회 이후의 트랜잭션을 탐지하고 예치된 채권을 소각(burn)합니다.

평판 영향**(Reputation Impact):** 슬래싱된 에이전트는 영구적인 평판 손상을 입습니다. 이후 인가를 받기 위해서는 더 높은 담보가 요구되며, 일부 서비스는 슬래싱 이력이 있는 에이전트를 완전히 거부할 수도 있습니다.

분배**(Distribution):** 소각된 자금은 피해 당사자에게 분배됩니다. 사용자는 오작동한 에이전트로 인한 손실을 복구하고, 서비스는 잘못된 요청 처리에 대한 보상을 받습니다.

이 경제적 레이어는 네트워크 및 암호학적 철회 메커니즘이 실패하더라도, 에이전트가 철회를 무시하지 못하도록 강력한 금전적 동기를 제공합니다. 철회를 무시할 때 발생하는 손실은 작동을 지속해 얻을 수 있는 이익보다 훨씬 크도록 설계되었습니다.

3.4.3.4 Graceful Degradation in Adversarial Conditions 적대적 환경에서의 점진적 복원력

Kite의 철회 시스템은 다양한 실패 모드에서 점진적으로 복원력 있게 작동하도록 설계되었습니다.

네트워크 분할**(Network Partition):** 로컬 네트워크 내에서는 철회가 독립적으로 지속되며, 암호학적 인증서는 최종적 일관성(Eventual Consistency)을 제공합니다.

허브 장애**(Hub Failure):** 중앙 조정 없이 피어 투 피어 전파가 지속되며, 서비스들은 직접 철회 정보를 교환합니다.

블록체인 혼잡**(Blockchain Congestion):** 오프체인 철회 메커니즘은 온체인 상태와 무관하게 독립적으로 작동합니다. 온체인 슬래싱은 궁극적으로 집행을 보장합니다.

서비스 오프라인**(Service Offline):** 캐싱된 철회 정보는 지속되며, 서비스가 재시작될 때 즉시 철회가 적용됩니다.

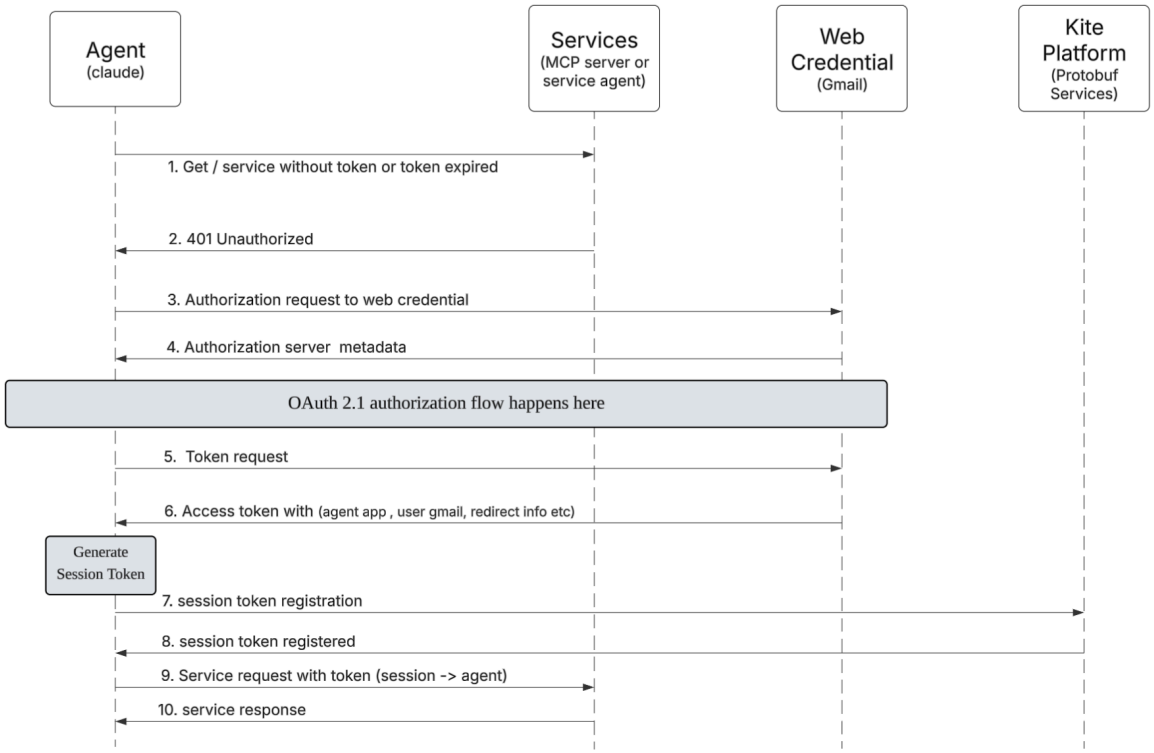
이 다층적 접근 방식은 적대적 조건이나 인프라 장애 상황에서도 철회 기능이 유지되도록 보장합니다. 사용자는 시스템 상태와 무관하게 항상 자신의 에이전트를 완전히 통제할 수 있습니다.

4. 에이전트 플로우 (Agent Flows)

에이전트의 상호작용 과정은 권한 부여 수립, 지속적 통신, 결제를 통한 가치 교환의 세 단계로 구성됩니다. 각 단계는 암호학적 신뢰를 기반으로 설계되어, 개발자에게는 직관적인 구축 경험을, 사용자에게는 완전한 통제권을 제공합니다.

4.1 에이전트 권한 부여 절차(Agent Authorization Flow)

권한 부여 절차는 전통적인 웹 인증과 블록체인 정산을 연결하는 정교한 흐름을 통해, 사용자의 아이덴티티를 에이전트의 실행 권한으로 전환합니다.



4.1.1 권한 부여의 과제 **(The Authorization Challenge)**

이 절차는 에이전트가 유효한 자격 증명 없이 서비스에 접근을 시도할 때 시작됩니다. 서비스는 401 Unauthorized 응답을 반환하며, 필요한 인증 방식을 명시합니다. 이 과정을 통해 단회성 사용자 인증이 지속 가능한 에이전트 권한으로 전환되는 권한 부여 시퀀스가 시작됩니다.

핵심 통찰(Key Insight): Gmail/OAuth는 사용자의 웹 아이덴티티를 증명합니다(예: `did:kite_chain_id:claude:scott` — Claude 애플리케이션과 Scott의 Gmail ID를 결합한 식별자). Kite 세션 토큰은 이 일회성 인증을 시간 제한과 정책 기반 보호가 적용된 형태로 변환하여, 사용자의 주요 자격 증명을 반복적으로 노출하지 않고도 에이전트가 안전하게 사용할 수 있는 권한을 제공합니다.

4.1.2 권한 부여 참여자 및 시퀀스 **(Authorization Actors and Sequence)**

권한 부여 절차에는 네 가지 주요 참여자가 있습니다.

- 에이전트(예: Claude): 서비스 접근을 요청하는 AI 시스템
- 서비스(예: MCP 서버 또는 서비스 에이전트): 접근 대상 리소스
- 웹 자격 증명 제공자(예: Gmail): 아이덴티티 검증 소스
- **Kite** 플랫폼 및 체인: 인가 및 정산 레이어

권한 부여 절차는 다음 여섯 단계로 진행됩니다.

1단계**:** 초기 요청 실패**(Initial Request Failure)**

에이전트가 유효하지 않거나 만료된 토큰으로 서비스를 호출하면, 서비스는 401 Unauthorized 응답을 반환하며 인가 프로세스가 시작됩니다.

2단계**:** 디스커버리 단계**(Discovery Phase)**

서비스는 웹 자격 증명이 필요함을 알리고 Gmail 또는 지원되는 다른 제공자를 지정합니다. 에이전트는 표준 디스커버리 메커니즘을 통해 권한 부여 서버의 메타데이터를 조회합니다.

3단계**: OAuth** 인증**(OAuth Authentication)**

표준 OAuth 2.1 흐름이 Gmail을 통해 수행됩니다. 사용자는 로그인하고 접근 권한을 승인합니다. 에이전트는 토큰 요청을 수행해, 에이전트 애플리케이션·사용자의 Gmail 아이덴티티·리디렉션 정보가 결합된 액세스 토큰을 발급받습니다. 이를 통해 실제 사용자가 특정 에이전트를 승인했다는 암호학적 증거가 생성됩니다.

4단계**:** 세션 토큰 등록**(Session Token Registration)**

에이전트는 웹 자격 증명을 획득한 후 로컬 세션 키를 생성하고 Kite 플랫폼 및 체인에 세션 토큰을 등록합니다. 이 등록에는 다음 정보가 바인딩됩니다.

- 에이전트 및 애플리케이션 아이덴티티

- 허용된 범위(Scopes)와 작업(Operation)
- 할당량(Quota) 및 TTL(Time to Live)
- 사용자 인가로 이어지는 증명 체인(Proof Chain)

세션 개인키는 로컬 환경을 벗어나지 않으며, DID 기반 세션 등록 호출을 통해 세션은 네트워크 전역에서 인식 가능한 상태로 연결 및 기록됩니다.

5단계**:** 서비스 재시도**(Service Retry)**

에이전트는 임시 세션 키(ephemeral session key)로 서명된 세션 토큰을 포함하여 원래 요청을 다시 보냅니다.

6단계**:** 검증 및 실행**(Verification and Execution)**

서비스는 토큰을 검증하고 이를 Kite의 레지스트리 및 정책과 대조합니다. 토큰이 유효하며, 정책 및 할당량 검사를 통과하고, 시간 제한 내에 있으면 서비스는 요청을 실행하고 성공 응답을 반환합니다.

4.1.3 JWT 토큰 구조

세션 키가 승인 및 생성되면 이에 상응하는 JWT 토큰이 발행되어 모든 인가 정보를 캡슐화합니다.

```
{
  "agent_did": "did:kite:alice.eth/chatgpt/assistant-v1",
  "apps": ["chatGPT", "cursor"],
  "timestamps": {
    "created_at": 1704067200,
    "expires_at": 1704070800
  },
  "proof_chain": "session→agent→user",
  "optional": {
    "user": "did:kite:alice.eth",
```

```
    "reputation_score": 850,
    "allowed_actions": ["read", "write", "pay"]
  }
}
```

JWT 토큰이 발급된 이후, 네트워크 내 모든 서비스 요청에는 JWT 토큰과 암호화용 세션 공개키가 함께 포함됩니다. 이러한 이중 인증 구조는 권한 검증과 데이터 진위성을 동시에 확보합니다.

4.2 에이전트 평판 및 신뢰 축적 (Agent Reputation and Trust Accumulation)

기존 블록체인은 모든 계정을 동등하게 취급합니다. 새로 생성된 주소도 수년간 정상적인 활동을 해온 계정과 동일한 권한을 가집니다. 이러한 평등주의적 접근은, 행동 이력이 인가 범위를 결정해야 하는 에이전트 시스템에서는 치명적인 결함으로 작용합니다.

4.2.1 에이전트 시스템의 신뢰 역학 (Trust Dynamics for Agent Systems)

에이전트 시스템은 기존 블록체인이 제공할 수 없는 정교한 신뢰 역학을 필요로 합니다.

점진적 인가**(Progressive Authorization):** 신규 에이전트는 하루 $10 한도나 제한된 서비스 접근과 같은 최소 권한에서 시작해야 합니다. 각 성공적인 작업이 평판 점수에 반영되어, 에이전트의 기능이 자동으로 확장됩니다.

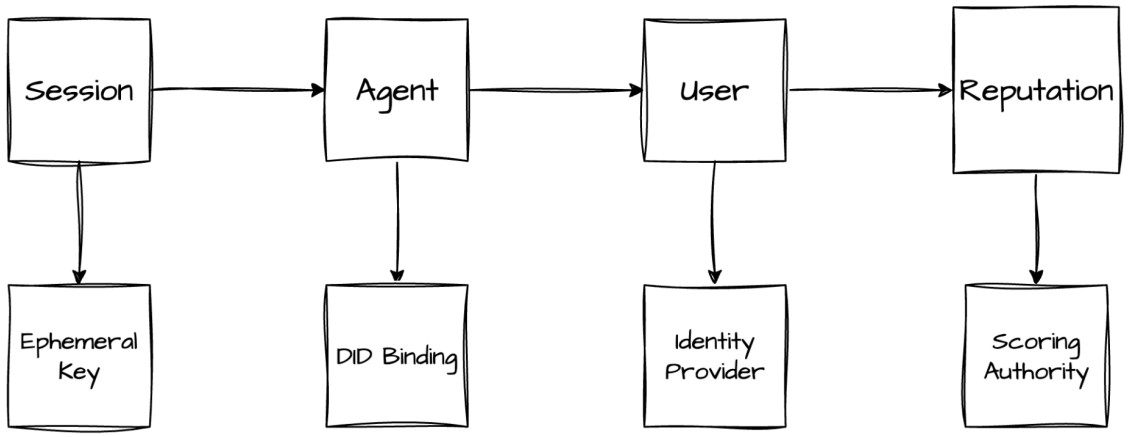
행동 기반 조정**(Behavioral Adjustment):** 성공적인 운영이 누적될수록 인가 한도는 점진적으로 증가해야 합니다. 신뢰가 축적된 에이전트는 $10 → $100 → $1,000으로 한도가 상승할 수 있습니다. 반대로 위반이 발생하면 시스템은 자동으로 제약을 강화하여 한도를 축소하거나 추가 검증을 요구합니다.

신뢰 이동성**(Trust Portability):** 신뢰는 서비스 간에 전이되어야 합니다. 한 플랫폼에서 평판을 쌓은 에이전트는 새로운 환경에서도 초기부터 신뢰를 부트스트래핑할 수 있어야 합니다.

검증 경제학**(Verification Economics):** 네이티브 평판 시스템이 없다면, 모든 에이전트 상호작용은 '제로 트러스트(Zero Trust)' 상태에서 시작되어 매번 검증 비용이 발생합니다. 이 오버헤드는 마이크로 결제를 경제적으로 불가능하게 만듭니다.

4.2.2 증명 체인 아키텍처 **(Proof Chain Architecture)**

Kite는 세션→ 에이전트→ 사용자 → 평판에 이르는 완전한 증명 체인을 제공합니다. 이 체인은 검증된 기관에 의해 서명되며, 반복적인 인증 절차 없이도 즉시 신뢰 검증이 가능합니다.



서비스 제공자와 사용자는 이 증명 체인을 활용하여, 무조건적인 신뢰가 아닌 검증 가능한 이력에 기반한 인가 결정을 내릴 수 있습니다. 예를 들어 사용자는 다음과 같은 기준으로 접근 권한을 부여할 수 있습니다.

- 평판 점수 100 이상: 읽기(Read) 권한
- 평판 점수 500 이상: 쓰기(Write) 권한
- 평판 점수 750 이상: 결제(Payment) 권한
- 평판 점수 900 이상: 무제한 접근(Full Access)

이러한 단계적 신뢰 모델은 보안을 유지하면서도 점진적 자율성(progressive autonomy)을 구현합니다.

4.3 에이전트 통신 흐름 **(Agent Communication Flow)**

인간의 트랜잭션은 개별적으로 발생하지만, 에이전트의 작동은 지속적인 통신과 협력을 전제로 합니다. 따라서 에이전트 환경에서는 지속 연결, 다자 간 협력, 검증 가능한 메시지 교환에 대한 네이티브 지원이 필수적입니다.

4.3.1 에이전트 간 메시징 **(Agent to Agent Messaging, A2A)**

암호화된 채널은 전략이나 개인 정보를 노출하지 않고도 협상, 탐색, 조율을 가능하게 합니다. A2A 프로토콜은 에이전트의 기능과 엔드포인트에 대한 진실의 원천 역할을 하는 Agent Card를 통해 구조화된 통신을 제공합니다.

Agent Card 구조:

```
AgentCard {
  agent_did: "did:kite:alice.eth/gpt/trader",
  capabilities: ["streaming", "push_notifications"],
  security_schemes: ["JWT", "session_key"],
  endpoints: {
    primary: "wss://agent.example.com",
    fallback: "https://agent.example.com/api"
  },
  auth_methods: ["oauth", "did_auth"],
  session_scheme: "JWT + session_public_key"
}
```

상대 노드는 Agent Card를 조회하여 해당 에이전트의 기능, 지원되는 보안 스킴(Security Scheme), 인증 방식을 파악합니다. 이 카드에는 엔드포인트 URL, 인증 방식, 사용 가능한 메서드가 명시되며, 가장 중요한 것은 에이전트 DID와 세션 보안 스킴이 포함되어 있어, A2A 호환 피어는 이를 기반으로 세션 범위 자격 증명을 검증할 수 있다는 점입니다.

4.3.2 서비스 수준 계약 (Service Level Agreements, SLAs)

- 응답 시간**(Response Time):** 서비스는 100ms 이내에 응답해야 하며, 이를 초과할 경우 자동 패널티가 부과됩니다.
- 가용성**(Availability):** 99.9%의 업타임을 유지해야 하며, 다운타임에 대해서는 비례 환불(pro-rata refund)이 적용됩니다.

- 정확도**(Accuracy):** 오류율이 0.1%를 초과하면 평판 슬래싱(reputation slashing)이 발생합니다.
- 처리량**(Throughput):** 초당 최소 1,000건의 요청을 보장합니다.

이러한 프로그래머블 SLA는 모호한 약속을 암호학적으로 집행 가능한 보증으로 전환합니다.

4.3.3 평판 네트워크 **(Reputation Networks)**

지속적인 아이덴티티 관리와 성과 기록은 서비스 간 신뢰를 축적하고 확장할 수 있는 기반을 제공합니다. 각 상호작용은 글로벌 평판 시스템에 반영되어 네트워크 전반의 신뢰 형성에 기여합니다.

- 성공적인 결제는 점수를 높입니다.
- 빠른 응답은 평판을 향상시킵니다.
- 실패한 전달은 신뢰를 낮춥니다.
- SLA 위반은 페널티를 유발합니다.

이렇게 구축된 평판은 네트워크 전반에 걸쳐 이동 가능하며, 새로운 관계가 매번 제로 트러스트(Zero Trust)에서 시작되는 콜드스타트 문제를 해결합니다.

4.4 에이전트 결제 흐름 **(Agent Payment Flow)**

기존 결제망은 사람이 승인하는 절차를 전제로 설계된 대규모·비정기 결제 구조로, 에이전트가 수행하는 마이크로결제 환경에는 근본적으로 부적합합니다. 월간 구독이나 1회 결제는 높은 고정 수수료, 느린 정산, 긴 차지백 기간, 최소 거래 단위 등의 구조로 인해 메시지 단위의 과금(pay-per-message)이 불가능합니다.

Kite는 이 모델을 완전히 뒤집습니다. 에이전트는 서비스와 상태 채널을 개설하고, 모든 상호작용마다 서명된 마이크로 바우처를 교환하며, 단 한 번의 채널 종료로 온체인에서 잔액을 정산합니다. 수수료는 수백만 건의 호출에 걸쳐 분산되며, 이를 통해 메시지당 약 $0.000001 수준의 진정한 마이크로 결제가 가능해집니다. 초당 또는 토큰 단위 스트리밍 결제, 자동 할당량, 즉시 철회 등이 스마트 계정과 세션 키에 의해 자동으로 집행됩니다. 결과적으로 결제는 청구 주기가 아닌 네트워크 패킷처럼 실시간으로 작동하는 사용량 기반 과금 모델로 진화합니다.

4.4.1 채널 개설 **(Channel Opening)**

스마트 계정은 일정 한도(예: $50)를 채널 컨트랙트에 예치하며 다음과 같이 기록됩니다.

```
{
```

```
  merchant_did: "did:kite:service.ai/api",
  session_did: "did:kite:alice.eth/gpt/session-xyz",
  expiry: 1704070800,
  challenge_window: 3600  // 1 hour dispute period
}
```

이 예치는 정상적인 운영을 담보하는 보증금 역할을 합니다. 양측은 서명된 상태 업데이트를 교환해 채널의 초기 상태를 설정하고, 이후에는 온체인 부담 없이 오프체인에서 신속하게 거래할 수 있습니다.

4.4.2 상호작용별 결제 (Pay Per Interaction, Off Chain)

각 메시지나 API 호출에는 세션 키로 서명된 바우처(Voucher)가 포함됩니다.

```
{
  channel_id: "0xchan...",
  cumulative_amount: "10.523",  // Running total in USDC
  nonce: 4567,                   // Monotonic counter
  timestamp: 1704068000,
  signature: "0xsig..."
}
```

채널은 이전 상태와 새 상태를 추적하며, 모든 참여자는 상태 변경에 합의해야 합니다. 상인은 바우처가 유효하고 할당량 및 TTL 내에 있으면 작업을 수락합니다. 스트리밍 결제는 $0.000001 단위의 작은 증분으로 흐르며, 지출 상한, 화이트리스트, 롤링 윈도우 정책이 바우처 생성 이전에 Kite 사이드카(Sidecar)에서 로컬로 강제됩니다.

4.4.3 채널 정산 (Channel Settlement)

채널은 협력적(Cooperative) 혹은 분쟁 기반(Disputed) 경로로 종료됩니다.

협력적 종료**(Cooperative Close, Happy Path):**

양측이 최종 상태에 공동 서명한 뒤 한 번만 제출합니다. 미사용 자금은 즉시 사용자에게 반환되며, 가스비는 최소화되고 정산은 즉시 완료됩니다.

분쟁 종료**(Disputed Close):**

한쪽 참여자가 가장 최근 서명된 상태를 블록체인에 게시합니다. 상대방은 지정된 기간 내에 더 최신 상태를 제출하여 도전할 수 있으며, 이를 통해 가장 최근의 업데이트만이 최종 정산됩니다. 합의 파기나 이상 상황이 발생하면, 어느 쪽이든 온체인 컨트랙트를 호출하여 채널을 강제로 종료하고 자금을 분배할 수 있습니다.

일반적인 분쟁 사례는 다음과 같습니다.

- 참여자가 오프라인으로 전환되어 상태 전이를 제안하지 않는 경우
- 유효한 업데이트에 공동 서명을 거부하는 경우
- 오래된 상태로 최종화를 시도하는 경우
- 유효하지 않은 상태 전이를 제안하는 경우

이러한 분쟁 해결 메커니즘은 완전한 비신뢰성을 보장합니다. 상대방의 행위와 무관하게 정직한 참여자는 언제든 자신의 예치를 회수할 수 있습니다. 블록체인은 궁극적인 중재자로 작동하며, 정직한 행동은 항상 보상받고 악의적 행동은 항상 처벌받도록 보장합니다.

4.4.4 경제적 전환 (Economic Transformation)

본 아키텍처는 기존 인프라로는 불가능했던 새로운 경제 모델을 가능하게 합니다.

세분화된 가격 책정 **(Granular Pricing):** 서비스는 개별 API 호출, Inference 토큰, 컴퓨팅 밀리초 단위까지 가격을 세분화할 수 있습니다. 예를 들어 하나의 대화는 언어 모델 추론에 $0.02, 임베딩 생성에 $0.001, 스토리지 사용에 $0.0001이 들 수 있습니다.

실시간 경제 **(Real Time Economics):** 가격은 부하, 우선순위, 자원 가용성에 따라 실시간으로 조정됩니다. 피크 시간에는 요금이 상승하고, 프리미엄 모델은 더 높은 요율을 가집니다. 긴급 요청은 우선 수수료(priority fee)를 지불합니다.

자동 예산 관리 **(Automatic Budgets):** 에이전트는 암호학적으로 집행되는 지출 한도 내에서 운영됩니다. 예산이 한도에 근접하면 에이전트는 자동으로 작동을 조절하거나 중단합니다.

즉시 정산 **(Instant Settlement):** 월별 청구나 정산 대기 시간이 없습니다. 모든 상호작용은 즉시 정산되며, 서비스는 즉각 대금을 수령하고 사용자는 실시간 지출을 확인합니다.

청구 주기에서 패킷 단위 경제로의 전환은 기존 결제 시스템에서는 불가능했던 비즈니스 모델을 열어줍니다. 모든 메시지는 결제가 되고, 모든 결제는 프로그래머블하며, 모든 프로그램은 검증 가능합니다. 이 것이 진정한 자율형 에이전트를 뒷받침하는 경제적 기반입니다.

5. 기존 접근 방식의 한계와 Kite의 차별성 (Comparison With Existing Approaches)

여러 프로젝트들이 에이전트 결제라는 복잡한 퍼즐의 일부를 해결하려 시도해왔지만, Kite는 자율형 에이전트 운영을 위해 제1 원리에 따라 설계된 최초의 완전한 인프라를 제공합니다. 기존 접근 방식을 어떻게 넘어섰는지를 이해하면, 왜 부분적인 해법으로는 에이전트 경제를 실현할 수 없는지가 명확해집니다.

5.1 Tempo를 넘어: 결제 중심에서 에이전트 네이티브로 (Beyond Tempo: From Payments-First to Agent-Native)

Tempo는 현재 블록체인 결제 최적화의 최신 수준을 대표하지만, 여전히 인간 중심 설계라는 근본적 한계를 가집니다. Tempo가 결제 효율성을 높였다면, Kite는 자율 운영에 필수적인 에이전트 네이티브 인프라를 제공합니다.

Tempo가 멈춘 곳에서 **Kite**가 시작합니다**.**

Tempo는 안정적인 수수료와 전용 레인을 활용해 인간이 시작하는 결제를 최적화합니다. 이는 중요한 문제를 해결하지만, 에이전트 시스템의 근본적 요구사항을 충족하지 못합니다. Kite는 그 한계를 넘어섭니다.

계층형 아이덴티티 아키텍처 **(Hierarchical Identity Architecture):** Tempo가 모든 참여자를 동등하게 취급하는 반면, Kite는 사용자 → 에이전트 → 세션으로 이어지는 3계층 암호학적 위임 구조를 구현합니다. 이는 기능 추가가 아니라, 자율 시스템에서 권한이 흐르는 방식을 근본적으로 재구성한 것입니다.

프로그래머블 제약 집행 **(Programmable Constraint Enforcement):** Tempo는 인간이 각 트랜잭션을 평가한다고 가정합니다. Kite는 에이전트가 오류나 환각(hallucination)을 일으킬 수 있음을 인정하고, 스마트 컨트랙트 기반의 제약을 통해 에이전트가 한계를 넘지 못하도록 수학적으로 보장합니다.

에이전트 네이티브 트랜잭션 타입 **(Native Agent Transaction Types):** Tempo는 결제만 처리하지만, Kite는 연산 요청, API 호출, 데이터 쿼리를 트랜잭션 내에 직접 내장합니다. 모든 메시지는 인가와 전달이 암호학적으로 증명된 과금 가능 이벤트가 됩니다.

프로토콜 수준 통합 **(Protocol-Level Integration):** Tempo가 고립된 상태로 남아있는 반면, Kite는 A2A, MCP, OAuth 2.1 및 신규 에이전트 표준들과 네이티브하게 호환됩니다. 에이전트는 추가 적응 없이 여러 생태계에서 동작할 수 있습니다.

Tempo는 블록체인 결제를 개선했지만, Kite는 에이전트 네이티브 인프라라는 새로운 패러다임을 제시했습니다.이는 단순한 기술적 발전이 아니라,설계 철학의 근본적 전환입니다.

5.2 탈중앙 아키텍처 **vs SkyFire**의 중앙화 **(Decentralized Architecture vs. SkyFire's Centralization)**

SkyFire pioneered agent payments but chose architectural patterns that create systemic vulnerabilities. Kite's decentralized design eliminates these critical weaknesses.

SkyFire는 에이전트 결제 분야의 초기 개척자였지만, 시스템 전반에 취약성을 초래하는 중앙화 아키텍처를 선택했습니다. 반면, Kite는 탈중앙 설계를 통해 이러한 구조적 약점을 제거하고, 신뢰 없는 환경에서도 안정적으로 작동하는 인프라를 구축했습니다.

5.2.1 단일 장애점 제거 **(Eliminating Single Points of Failure)**

SkyFire의 치명적 결함**:** 서버 측 JWT 발급과 중앙화된 키 관리 시스템은 시스템 전반에 치명적인 위험을 초래합니다. 단 한 번의 침해로 모든 에이전트가 노출되고, 한 건의 법적 조치로 전체 운영이 중단될 수 있으며, 특정 기업의 실패가 생태계 전체의 붕괴로 이어질 수 있습니다.

Kite의 복원력 있는 아키텍처**:** Kite는 클라이언트 측 토큰 생성과 임시 세션 키를 통해 단일 장애점을 원천적으로 제거했습니다. 따라서 설령 Kite가 내일 운영을 중단하더라도, 에이전트는 자율적으로 작동을 계속할 수 있습니다. 이는 단순한 보안 개선이 아니라, 존립 리스크 자체를 제거한 구조적 혁신입니다.

5.2.2 중개 없는 신뢰 **(Trust Without Intermediaries)**

SkyFire의 의존성**:** SkyFire의 서비스 구조에서는, 각 서비스가 에이전트를 신뢰하기 위해 먼저 SkyFire 자체를 신뢰해야 합니다. 이로 인해 플랫폼 종속성과 구조적 리스크가 발생하며, 결과적으로 SkyFire가 모든 신뢰 결정의 병목으로 작동합니다.

Kite의 수학적 신뢰**(Mathematical Trust):** Kite는 세션–에이전트–사용자로 이어지는 완전한 증명 체인을 구축하고, Gmail·Twitter 등 기존 신뢰 기반 제공자를 루트로 삼습니다. 이를 통해 서비스는 Kite를 신뢰하지 않고도 에이전트를 암호학적으로 검증할 수 있습니다. Kite는 신뢰를 매개하지 않는 중립적 인프라로 기능하며, 시스템 전반에서 수학적으로 증명 가능한 신뢰 구조를 제공합니다.

5.2.3 유연한 결제 아키텍처 **(Flexible Payment Architecture)**

SkyFire의 경직된 결합**(SkyFire's Rigid Coupling):** SkyFire는 결제 정보를 JWT 토큰에 직접 포함시켜 다양한 결제 패턴을 제한하고, 복합적인 사용 사례를 복잡하게 만듭니다.

Kite의 모듈형 설계**(Kite's Modular Design):** Kite는 인증(Authentication)과 결제(Payment) 흐름을 완전히 분리하여, 스트리밍 결제(streaming payments), 조건부 에스크로(conditional escrow), 구독 모델(subscription models), 그리고 향후 혁신적 결제 방식을 핵심 인프라 변경 없이 구현할 수 있습니다.

5.3 네이티브 상호운용성 **vs** 분절된 표준 **(Native Interoperability vs Fragmented Standards)**

Agent Commerce Kit(ACK), Google의 AP2, ERC-8004 등 여러 프로젝트가 유용한 표준을 정의하고 있지만, 이러한 비전을 실제로 작동시키는 실행 레이어(execution layer) 가 부재합니다. Kite는 단순히 이 표준들을 준수하는 수준이 아니라, 그것들이 실제로 작동하도록 하는 인프라 자체를 제공합니다.

5.3.1 ACK 의 운영화 **(Making ACK Operational)**

ACK는 에이전트 상거래(agent commerce)의 원칙을 정의하지만, Kite는 이를 실제로 구현합니다.

- 에이전트 패스포트**(Agent Passport):** 단순한 아이덴티티가 아니라 선택적 공개(selective disclosure)를 포함한 완전한 자격 관리 시스템
- 프로그래머블 결제 레인**(Programmable Payment Lanes):** 단순 결제가 아니라 에이전트 네이티브 트랜잭션 타입
- 의도 기반 인가**(Intent-Based Authorization):** 단순한 정책이 아닌, 암호학적으로 집행되는 제약 조건

Kite는 ACK의 명세(specification)를 작동하는 인프라(working infrastructure) 로 전환합니다.

5.3.2 Google AP2 의 확장 **(Extending Google's AP2)**

AP2는 에이전트 결제를 위한 인텐트 명령(Intent Mandate)을 도입했지만, Kite는 이를 넘어섭니다.

- 3계층 아이덴티티 구조(Three-Tier Identity Hierarchy)를 통한 정밀한 권한 위임
- AP2가 달성하지 못한 마이크로 스트리밍 결제를 가능하게 하는 상태 채널(State Channels)
- 단순 명령을 넘어 서비스 보증을 암호학적으로 집행하는 SLA 계약

5.3.3 ERC-8004 비전의 구현 **(Implementing ERC-8004's Vision)**

ERC-8004는 아이덴티티, 평판, 검증을 위한 레지스트리를 제안합니다. Kite는 이를 다음과 같이 구현합니다.

- 아이덴티티 요구사항을 초과 충족하는 검증 가능한 자격 증명(VC)을 포함한 에이전트 패스포트
- 모든 상호작용에 걸쳐 누적되는 행동 기반 평판 시스템 (Behavioral Reputation)
- 반박 불가능한 검증 결과를 제공하는 암호학적 증명 체인(Cryptographic Proof Chain)

5.4 완전한 해결책 **(The Complete Solution)**

다른 프로젝트들은 문제의 조각만을 다룹니다. Tempo는 결제 효율성을 개선했지만 에이전트의 존재를 고려하지 않았고, SkyFire는 에이전트를 가능하게 했지만 구조적 위험을 초래했습니다. 각종 표준은 인터페이스만 제시할 뿐, 실제 구현이 결여되어 있습니다.

Kite는 이 모든 요소를 하나의 통합된 스택으로 제공합니다. 에이전트 패턴에 최적화된 블록체인부터 위임 구조를 지원하는 아이덴티티 시스템, 모델 환각이나 오류를 방지하는 프로그래머블 제약, 마이크로페이먼트 스트리밍을 구현하는 상태 채널, 네이티브 프로토콜 호환성, 그리고 암호학적으로 검증된 보안 보증에 이르기까지 전 계층을 아우릅니다.

Kite가 만든 것은 단순한 결제 시스템이 아닙니다. 우리는 에이전트 경제를 실현하는 기반 인프라를 구축했습니다.

6. 활용 사례

에이전트 자율성과 프로그래머블 결제의 결합은 기존 인프라로는 구현할 수 없었던 새로운 비즈니스 모델을 열었습니다. 다음 사례들은 모든 상호작용이 거래로 전환될 때, 그 변화가 만들어내는 구조적 잠재력을 보여줍니다.

6.1 게임 산업**:** 글로벌 규모의 진정한 마이크로 결제 **(Gaming: True Microtransactions at Global Scale)**

기존 구조**:** $0.50짜리 스킨이나 $1짜리 추가 생명을 구매하더라도, 신용카드 결제망은 $0.30의 고정 수수료와 2.9%의 비율 수수료를 부과하며, 정산까지 수일이 걸리고 환전 수수료도 발생합니다. 이러한 구조에서는 소액 결제가 경제적으로 불가능합니다.

Kite의 에이전트 네이티브 결제망**(Agent-Native Rails)** 을 활용하면, 플레이어는 단 한 번의 클릭으로 즉시 PYUSD 결제를 수행합니다. 카드 토큰화도, 은행 중개자도 필요 없이 암호학적 가치 전송만으로 거래가 완료됩니다. 브라질의 게이머가 미국 게임 스튜디오로부터 환율 수수료나 결제 실패 없이 상품을 구매할 수 있습니다.

게임 에이전트는 여러 인앱 구매를 자동으로 묶어 처리하고, 부모가 설정한 지출 한도를 프로그램적으로 강제하며, 로열티 크레딧을 자동으로 적용합니다. 예를 들어 “월 최대 $20” 한도를 설정하면 블록체인이 이를 절대적으로 집행합니다. 게임 개발자는 즉시 정산을 받아 크리에이터와의 수익 분배를 실시간으로 처리할 수 있습니다.

이 구조는 기존에는 비경제적이었던 1달러 미만의 상품 판매를 가능하게 만듭니다. 게임은 결제 한도를 맞추기 위한 번들링이 아닌 실제 가치에 기반한 가격 책정이 가능해집니다. 게임 경제 전반이 더욱 유동적이고 투명하며 접근 가능하게 변합니다.

6.2 IoT: 머신 간 대역폭 시장 (Machine-to-Machine Bandwidth Markets)

기존 구조: IoT 기기는 월 단위 요금제나 고정 요율 기반의 통신망에 의존합니다. 5분간만 대역폭이 필요한 드론도 한 달 요금을 지불해야 합니다.

Kite의 에이전틱 결제(Agentic Payments)를 통해, 디바이스는 ISP나 위성 제공자에게 초 단위 PYUSD 결제 채널을 개설합니다. 각 데이터 패킷이 소비될 때마다 스트리밍 마이크로 결제가 자동으로 발생하며, 세션은 즉시 정산됩니다. 차량은 지도 업데이트에 $0.02를, 센서는 텔레메트리 업로드에 $0.001을, 드론은 영상 스트리밍에 분당 $0.50을 지불합니다.

글로벌 스테이블코인 결제망을 통한 청산은 통신사 간 환전 및 로밍 복잡성을 제거합니다. 기기는 전 세계 어디서든 네트워크에 연결해 실제 사용량만큼만 비용을 지불할 수 있습니다. 결과적으로 기계 간 실시간 과금형 대역폭 시장(pay-as-you-go bandwidth market)이 가능해져 IoT 배포 비용은 급격히 낮아지고 확장성은 크게 향상됩니다.

6.3 크리에이터 이코노미: 팬과 창작자 간 직접 수익 (Creator Economy: Direct Fan-to-Creator Revenue)

기존 구조: 크리에이터의 수익은 광고 네트워크나 플랫폼을 거치며 30%의 수수료가 공제됩니다. 최소 출금액($5~$10)과 수주일의 정산 대기 기간은 참여 장벽을 높입니다.

Kite의 스테이블코인 네이티브 결제(Stablecoin-Native Payments)에서는 팬이 실시간으로 센트 단위 이하의 PYUSD 팁을 보낼 수 있습니다. 예를 들어 시청자가 스트리밍을 보는 동안 10초마다 $0.01을 지불하며, 지속적인 수익 흐름을 형성합니다. 에이전트는 마이크로 팁을 자동 집계하여 “크리에이터 70% / 플랫폼 20% / 모더레이터 10%”로 투명하고 프로그래머블하게 분배합니다. 최소 금액도, 대기 기간도 없습니다. 단지 즉시 전송되는 가치만 존재합니다.

크리에이터는 은행 계좌 없이도 전 세계에서 결제를 수령할 수 있습니다. 나이지리아의 뮤지션이 일본의 팬으로부터 국제 송금 수수료 없이 직접 수익을 얻을 수 있습니다. 이는 팬과 창작자 간 관계를 재정의하며, 월별 정산이 아닌 지속적이고 실시간적인 수익 흐름을 창출합니다.

6.4 API 경제: 모든 호출이 트랜잭션이 되는 구조 (API Economy: Every Call Becomes a Transaction)

기존 구조**:** API 과금은 월 단위 인보이스나 선불 크레딧 형태로 이루어집니다. 불투명한 가격, 수동 정산, 신용 리스크가 비효율을 야기합니다.

Kite의 에이전트 간 결제(Agent-to-Agent Billing)를 통해 각 API 호출은 실시간으로 미세한 단위의 결제 흐름으로 측정됩니다. LLM 엔드포인트는 토큰당 $0.00001, 날씨 API는 요청당 $0.0001, GPU 클러스터는 초당 $0.01의 비용을 청구할 수 있습니다.

에이전트는 상태 채널(State Channel)을 개설하고, 호출마다 마이크로 바우처(Micro-Voucher)에 서명하며, 세션 단위로 정산합니다. 처리량은 그대로 유지되면서 수수료는 사실상 0에 수렴합니다. 사용 내역은 암호학적으로 감사 가능하며, 공급자는 실제 사용량을 증명하고 클라이언트는 자동으로 한도를 집행합니다.

이 모델은 모든 API 호출을 자율적이고 자동 청산(Self-Clearing) 가능한 금융 트랜잭션으로 전환합니다. 인보이스도, 추심도, 신용 리스크도 없습니다. 단지 기계 속도의 가치 교환만 존재합니다.

6.5 전자상거래**:** 프로그래머블 결제 및 에스크로 **(E-Commerce: Programmable Checkout and Escrow)**

기존 구조**:** 전자상거래는 신용카드 인가, 지연된 정산, 높은 차지백 비용을 동반합니다. 구매자는 구제 수단이 부족하고, 판매자는 사기 위험에 노출됩니다.

Kite의 에이전트 기반 에스크로(Agentic Escrow)에서는 쇼핑 에이전트가 상점으로 직접 결제하는 대신, PYUSD를 온체인 에스크로에 잠급니다. 자금은 검증자(배송 오라클, SLA 에이전트 등)가 “배송 완료”, “서비스 수행 완료”, 또는 “분쟁 없이 타임아웃”을 확인할 때에만 해제됩니다.

부분 결제(Partial Capture)는 상인이 배송된 품목만 정산할 수 있게 하며, 환불 및 취소는 프로그램적으로 즉시 실행됩니다. 플랫폼 수수료는 캡처 시 자동 분배됩니다. 그 결과, 차지백이 존재하지 않는 프로그래머블 체크아웃 환경이 만들어지며, 에이전트가 비즈니스 규칙을 집행하고 사용자는 암호학적 보증을 기반으로 구제 수단을 확보합니다.

6.6 개인 금융**:** 자율적 자산 관리 **(Personal Finance: Autonomous Money Management)**

기존 구조**:** 금융 관리를 위해 사용자는 은행 비밀번호를 서드파티 앱에 공유하거나 거래 내역을 수동 입력해야 합니다. 자동화는 제한적이며 보안 위험이 큽니다.

Kite는 통제된 위임**(Controlled Delegation)** 모델을 통해 사용자가 예산 관리, 청구서 납부, 소액 투자까지 담당하는 AI 에이전트를 직접 운영할 수 있도록 지원합니다. 에이전트는 은행 비밀번호 대신, 지출 한도가 설정된 온체인 자격 증명을 보유합니다. 이를 통해 사용자의 자금은 안전하게 제어되며, 에이전트는 공공요금을 서비스 에이전트에 직접 납부하고, 절약 계좌로 마이크로 예금을 스트리밍하며, 매일 1달러씩 스테이블코인을 ETF에 분산 투자합니다.

모든 거래 내역(예: “전기요금 50달러 납부,” “ETF XYZ 20달러 매수”)은 암호학적으로 서명되어 온체인에 기록됩니다. 사용자는 즉시 감사를 수행할 수 있으며, 세금 정산 과정 또한 자동화됩니다. 결국, 에이전트는 프로그래머블 안전장치와 마이크로결제 기능을 갖춘 신뢰 가능한 자산 관리자로 진화하며, 기존 금융 기관이 제공하지 못하는 수준의 완전한 자동화 금융 운영을 실현합니다.

6.7 지식 시장: 탈중앙화된 전문성 (Knowledge Markets: Decentralized Expertise)

기존 구조: 지식 요청은 단일 API를 통해 처리되며, 가격은 불투명하고 품질은 보장되지 않습니다. 전문성은 여전히 폐쇄된 플랫폼 안에 머물러 있으며, 정당한 보상 체계가 부재합니다.

Kite의 협력형 네트워크(Collaborative Networks)에서는 사용자가 법률, 의학, 연구 등 다양한 영역의 복합적인 지식 요청을 제출하면, 전문 에이전트들이 협업하여 답변을 생성합니다. 각 기여는 그 가치에 따라 평가되고, 이에 비례한 마이크로결제가 실시간으로 분배됩니다. 예를 들어 법률 에이전트는 계약 해석으로 0.50달러, 의료 에이전트는 증상 분석으로 0.30달러, 연구 에이전트는 인용 검증으로 0.20달러를 받을 수 있습니다.

모든 아이덴티티, 기여, 보상은 온체인에서 검증되어 투명한 평판 구조를 형성합니다. 사용자는 불투명한 API 기반 서비스 대신 탈중앙화된 집단 전문성을 얻게 되며, 기여자들은 중개자 없이 공정하고 세분화된 보상을 받을 수 있습니다.

6.8 공급망: 자율형 상거래 네트워크 (Supply Chain: Autonomous Commercial Networks)

기존 구조: 공급망은 EDI, 수동 승인, 문서 기반 절차에 의존하며 정산까지 30~90일이 소요됩니다. 신뢰 구축에는 높은 검증 비용이 필요합니다.

Kite의 엔터프라이즈 에이전트**(Enterprise Agents)** 를 활용하면 제조사, 공급사, 운송사가 지출 한도가 명시된 에이전트를 배포해 자동으로 거래를 수행합니다. 재고가 부족해지면 에이전트가 주문을 협상하고, 결제를 에스크로에 예치하며, 물류를 자동 조정합니다. IoT 오라클은 “항구 도착,” “검수 통과” 등의 마일스톤을 확인하고, 조건부 자금 해제(conditional fund release)를 트리거합니다.

그 결과, 24/7 자율적으로 작동하는 공급망이 완성됩니다. 즉시 정산, 프로그래머블 에스크로, 위·변조 불가능한 감사 로그가 보장되며, EDI 프로토콜도, 수동 승인도, 신뢰 리스크도 사라집니다. 전 세계 상거래가 소프트웨어의 속도로 운영됩니다.

6.9 탈중앙 거버넌스: AI로 강화된 DAO (Decentralized Governance: AI-Enhanced DAOs)

기존 구조**:** DAO는 투표 참여 저조, 느린 의사결정, 비효율적인 멀티시그 운영 문제에 시달리고 있습니다.

Kite의 AI 거버너(AI Governors)는 DAO의 금고 운영—리밸런싱, 투표, 반복 거래 등—을 담당하며, 모든 행위는 온체인 정책 한도 내에서 집행됩니다. 에이전트는 시장 변동에 실시간으로 대응하지만, 주요 결정은 여전히 인간 토큰 보유자의 승인 하에 이루어집니다. 예를 들어, 에이전트는 자율적으로 5% 범위 내에서 금고 자산을 리밸런싱할 수 있지만, 더 큰 조정에는 DAO 투표 승인이 필요합니다.

모든 거래, 투표, 행위는 온체인에 투명하게 기록됩니다. DAO는 인간의 거버넌스와 AI의 실행력이 결합된 진정한 자율 조직으로 진화합니다. 멀티시그보다 빠르고, 블랙박스 봇보다 안전하며, 암호학적 증명을 통해 완전한 책임성을 보장합니다.

7. 향후 과제 (Future Work)

Kite는 결제 인프라를 넘어 하나의 에코시스템으로 진화하고 있습니다. 향후 연구·개발 단계에서는 프로그래머블 트러스트와 에이전트 네이티브 기능의 적용 범위를 결제에서 검증 가능한 연산과 자율적 조정의 영역으로 확장할 예정입니다.

7.1 영지식 검증 에이전트 자격 증명 (Zero-Knowledge Verified Agent Credentials)

Kite는 에이전트가 기본 데이터를 공개하지 않고 자신이 가진 속성을 증명할 수 있도록, 영지식 기반 검증 자격 증명(ZK-Verified Credentials) 전용 스토리지 모듈을 개발 중입니다. 이는 단순한 아이덴티티 확인을 넘어, 능력·자격·권한 등 복합적 인증 구조까지 확장됩니다.

ZK 검증 속성**(ZK-Verified Attributes):** 에이전트는 개인 정보를 노출하지 않고 나이, KYC 상태, 전문 자격증, 보안 인가 등을 증명할 수 있습니다. 예를 들어 의료 에이전트는 의사의 신원을 공개하지 않고도 자신이 공인 전문의 자격을 보유하고 있음을 입증할 수 있습니다.

맞춤형 스토리지 아키텍처**(Custom Storage Architecture):** 암호화된 자격 증명 저장소는 임시 복호화 키를 사용하여, 저장소가 침해되더라도 민감한 데이터가 보호되도록 설계됩니다. 자격 증명은 자동 만료되며, 주기적으로 갱신되고 필요 시 즉시 철회됩니다.

조합형 증명**(Composable Proofs):** 여러 자격 증명을 결합해 하나의 복합 증명을 생성할 수 있습니다. 예를 들어 에이전트는 “18세 이상이며 제재 대상이 아닌 공인 투자자”임을 단일 영지식 증명으로 입증할 수 있습니다.

7.2 검증 가능한 추론 및 연산 (Verifiable Inference and Computation)

Kite는 에이전트의 의사결정 결과뿐 아니라, 해당 결정이 어떤 모델 파라미터·프롬프트·데이터 소스를 기반으로 도출되었는지를 암호학적으로 증명하는 검증 가능한 추론 프리미티브를 도입할 예정입니다.

추론 검증**(Inference Verification):** 스마트 컨트랙트는 특정 입력과 특정 모델 가중치로부터 생성된 결과가 실제 해당 모델에서 도출된 것임을 검증합니다. 이를 통해 AI 의사결정에 대한 규제 적합성을 위한 암호학적 증거가 확보됩니다.

작업 증명**(Task Attestation):** 복합적인 다단계 에이전트 작업은 각 의사결정 단계를 보여주는 증명 체인을 생성합니다. 이에 따라 분쟁은 증언이 아니라 암호학적 증거를 기반으로 해결됩니다.

데이터 가용성 레이어**(Data Availability Layer):** 데이터 가용성 프로토콜을 활용해 추론 증명을 단기 저장함으로써, 블록체인의 장기 부하 없이 저비용으로 보존할 수 있습니다. 이러한 증명은 규제 요구 기간 동안 조회할 수 있는 상태로 유지되며, 이후 자동으로 만료됩니다.

7.3 이동 가능한 평판 네트워크 (Portable Reputation Networks)

에이전트와 서비스는 거래 이력, SLA 성과, 피어 검증에 기반해 온체인 상에서 이동 가능한 평판(Portable Reputation) 을 축적합니다. 이 평판은 Kite 생태계 전반에서 작동하는 핵심 프리미티브로 기능합니다.

행동 기반 점수화**(Behavioral Scoring):** 모든 상호작용은 평판 점수에 반영됩니다. 성공적인 결제는 신뢰를 높이고, 빠른 응답은 평점을 향상시키며, 실패한 전달은 점수를 낮춥니다. 블록체인은 이 모든 이력을 변조 불가능한 형태로 기록합니다.

크로스 플랫폼 이동성**(Cross-Platform Portability):** 하나의 플랫폼에서 획득한 평판은 암호학적 증명으로 다른 플랫폼으로 이전됩니다. Kite에서 높은 평판을 쌓은 에이전트는 타 플랫폼에서도 우대 조건을 받습니다.

자동화된 신뢰 결정**(Automated Trust Decisions):** 높은 평판을 가진 에이전트는 자동으로 더 높은 한도, 낮은 수수료, 프리미엄 접근 권한을 부여받으며, 악의적 행위자는 즉시 제한됩니다. 이렇게 신뢰는 코드로 정의되고, 상황 변화에 따라 실시간으로 반응하는 프로그래머블한 개념이 됩니다.

7.4 검증 가능한 서비스 검색 (Verifiable Service Discovery)

Kite는 중앙 디렉토리에 의존하지 않고 검증 가능한 증명을 통해 서비스 간 신뢰를 형성하는 비신뢰형(discovery) 레이어를 구축할 예정입니다.

기능 증명**(Capability Attestations):** 서비스는 암호학적 자격 증명을 통해 자신의 기능을 증명합니다. 예를 들어 LLM 서비스는 모델 크기, 학습 데이터, 성능 지표를 영지식 증명으로 제시할 수 있습니다.

규제 준수 증명**(Compliance Proofs):** 서비스는 영지식 증명을 통해 규제 적합성을 입증합니다. 금융 서비스는 기업 정보를 공개하지 않고도 SEC 등록 여부를 증명할 수 있습니다.

성능 이력**(Performance History):** 과거 SLA 이력은 온체인에서 조회 가능합니다. 에이전트는 서비스가 주장하는 기능뿐 아니라 실제 제공한 성과를 기반으로 신뢰 여부를 결정합니다.

7.5 포괄적 추적성과 증명 (Comprehensive Traceability and Attestation)

Kite는 다자간 복합 상호작용의 모든 에이전트 행위가 관련 참여자의 서명 증명과 함께 기록되도록 추적성 레이어를 확장합니다.

암호학적 계보**(Cryptographic Lineage):** 모든 결제, 메시지, 추론은 완전한 이력에 연결됩니다. 감사인은 자금의 출처부터 목적지까지를 수학적으로 추적할 수 있습니다.

규제 준수**(Regulatory Compliance):** 검증 가능한 기록은 경쟁 정보 노출 없이 규제 요건을 충족합니다. 금융 당국은 문서 요청 대신 암호학적 증거를 통해 규제 준수 여부를 검증합니다.

자동화된 구제**(Automated Recourse):** 분쟁은 자동으로 검증 가능한 이벤트 체인에 연결되며, 스마트 컨트랙트가 암호학적 증거를 기반으로 중재를 수행합니다. 해결 과정은 법적 절차가 아닌 프로그램적 절차로 이루어집니다.

이러한 연구와 개발을 통해 Kite는 결제 인프라를 넘어, 검증 가능한 자율 연산 플랫폼으로 진화합니다. 에이전트는 단순히 거래하는 수준을 넘어, 자신의 결정을 증명하고, 신뢰를 축적하며, 완전한 투명성 속에서 작동하는 새로운 경제 주체가 될 것입니다.

7.6 향후 제품 로드맵(Product Roadmap Ahead)

Kite는 향후 4개 분기 동안 블록체인 프로토콜, 인프라, 애플리케이션 전 영역에 걸쳐 제품 로드맵을 단계적으로 확장할 계획입니다. 아래는 향후 1년간의 핵심 개발 계획을 요약한 것입니다.

2025 Q4

블록체인 프로토콜

- 알파 메인넷(Alpha Mainnet) 출시
    - USDC 지원
    - 온·오프램프(On/Off-Ramp) 구축
    - 브릿지(Bridge) 통합

인프라

- 아이덴티티 및 인증 시스템
- 커스터디 TEE + HSM을 활용한 보안 키 커스터디 기반의 3계층 에이전트 ID 모델(사용자 / 에이전트 / 세션·태스크) 구축
- Kite 패스포트 MVP 출시(소셜 로그인 + DID 연동 + 검증 가능한 자격 증명)
- 증명 체인(Proof Chain) 및 감사 추적(Audit Trail) 시스템(에이전트 → 세션 → 액션 → 정산) 공개
- JWT 토큰을 OAuth 2.1 및 A2A 표준과 통합
- AA 지갑 출시(세션 기반 사전 승인)
- 에이전트 통신 채널(MCP)
- 네트워크 내 결제 채널(Agent Payment Channel)
- x402 호환성 확보

애플리케이션

- 쇼핑 에이전트 샌드박스(Shopping Agent Sandbox) 구축

2026 Q1

블록체인 프로토콜

- 퍼블릭 메인넷(Public Mainnet) 출시
    - PYUSD, USDT, RLUSD 지원
- 기본 프로그래머블 결제(pay-per-use, 스트리밍) 활성화
- Base 등 소비자 중심 체인과의 크로스체인 유동성 확대

인프라

- 에이전트 통신 채널(A2A)
- 외부 결제 채널(External Agent Payment Channel)
- 퍼블릭 릴리스를 위한 품질 개선
- 정책 거버넌스 API(프로그래머블 지출 한도, SLA 규칙)

애플리케이션

- 포트폴리오 트레이딩 및 최적화 에이전트

2026 Q2

블록체인 프로토콜

- 패스트 파이널리티(Fast Finality) 및 에이전트 네이티브 워크로드 최적화
- Non-EVM 체인(Sui, Solana)과의 크로스체인 상호운용성 구현

인프라

- 에이전트 디스커버리
- AI 추론 및 에이전트 통신용 상태 채널을 위한 커스텀 트랜잭션 레인 구축
- A2A, MCP, OAuth 2.1표준과의 프로토콜 수준 호환성 확장

애플리케이션

- 결제 레일(Payment Rails) 고도화

2026 Q3

인프라

- 에이전트 네이티브 스테이블코인 전용 결제 레인 구축
- 에이전트 평판 시스템(거래 및 상호작용 이력을 기반으로 한 랭킹·검색·평판 API)
- 에이전트 마켓플레이스
- SLA 계약 시스템
- 규제 및 컴플라이언스 파트너를 위한 감사 및 추적 API

애플리케이션

- 서비스 결제 및 소싱 결제 기능 구현

8. 결론

에이전트 경제는 먼 미래의 가능성이 아니라, 단지 인프라의 제약으로 인해 지연된 즉각적인 현실입니다. 언어 모델은 이미 복잡한 워크플로우를 관리하고, 정교한 의사결정을 수행하며, 시스템 간 조율까지 가능할 만큼 발전했습니다. 기업은 파일럿 단계를 넘어 상용 환경에 배포를 시작했고, 규제 기관은 알고리즘 책임성을 위한 명확한 프레임워크를 마련했습니다. 이제 남은 마지막 조각은 단 하나입니다 — 에이전트를 1급 경제 주체(First-Class Economic Actor) 로 대우할 수 있는 인프라입니다.

8.1 인프라의 필연성 (The Infrastructure Imperative)

기존 시스템은 모든 계층에서 에이전트를 지원하지 못합니다. 인증 구조는 인간 사용자를 전제로 설계되어, 에이전트를 일회성 로그인을 위한 OAuth 흐름에 종속시킵니다. 결제망은 0.01달러의 트랜잭션에도 최소 0.30달러의 수수료를 부과해 마이크로결제를 경제적으로 불가능하게 만듭니다. 신뢰 모델은 무제한 접근과 전면 차단 사이의 이분법적 선택만 제공합니다. 이것은 수정 가능한 결함이 아니라 아키텍처 수준의 불일치입니다.

점진적 개선으로는 이 간극을 메울 수 없습니다. API 키를 추가한다고 자격 증명 폭발(Credential Explosion) 문제가 해결되지 않으며, 결제 수수료를 낮춘다고 스트리밍 결제가 가능해지지 않습니다. 감사 로그를 개선한다고 해서 컴플라이언스의 암호학적 증거가 생기는 것도 아닙니다. 에이전트 경제는 제1 원리에서 다시 설계된 인프라를 필요로 합니다.

8.2 Kite의 종합적 해법 (Kite's Comprehensive Solution)

Kite는 자율형 에이전트 운영을 위한 완전한 인프라 스택을 제공합니다.

3계층 아이덴티티 구조 **(Three-Tier Identity):** 사용자–에이전트–세션으로 이어지는 계층형 암호학적 위임 체계는 권한이 인간에서 에이전트로, 그리고 개별 작업으로 안전하게 이전되도록 보장합니다. 각 계층은 수학적으로 증명 가능한 한계 내에서 자율적으로 작동합니다.

프로그래머블 제약 **(Programmable Constraints):** 스마트 컨트랙트는 에이전트가 환각이나 오류, 침해 상황에서도 지출 한도·시간 범위·운영 경계를 초과하지 못하도록 강제합니다. 코드가 곧 법이 됩니다.

네이티브 결제 프리미티브 (Native Payment Primitives): 상태 채널은 메시지당 0.000001달러 수준의 마이크로결제를 가능하게 하며, 전용 결제 레인은 예측 가능한 비용 구조를 제공합니다. 모든 상호작용은 즉시 정산되는 과금 가능한 이벤트로 전환됩니다.

범용 상호운용성 **(Universal Interoperability):** A2A, MCP, OAuth 2.1 등 주요 에이전트 표준과 네이티브 호환성을 갖춰, 에이전트는 생태계를 넘나들며 별도의 적응 과정 없이 작동할 수 있습니다. 하나의 인프라로 전 세계적 접근성이 확보됩니다.

암호학적 보안 **(Cryptographic Security):** 3중 서명 검증, 손실 한정 정리, 다계층 철회 메커니즘은 시스템의 모든 동작을 수학적으로 보장합니다. 신뢰는 약속이 아닌 검증 가능한 상태로 전환됩니다.

8.3 경제적 전환 The Economic Transformation

이 변화는 기술적 인프라를 넘어 경제 패러다임의 전환을 의미합니다. 모든 API 호출이 거래로, 모든 상호작용이 실시간 결제로 전환되면 비즈니스 모델 자체가 변합니다. 결제 수수료가 사실상 사라지면 완전히 새로운 경제가 열립니다. 에이전트가 자율적으로 자금을 관리하게 되면, 인간의 생산성은 기하급수적으로 증가합니다.

게임 경제에서는 1센트 단위 아이템이 활성화되고, IoT 기기는 패킷 단위로 대역폭 비용을 지불합니다. 크리에이터는 실시간 스트리밍 수익을 얻으며, 지식 시장은 기여도를 세분화해 보상하고, 공급망은 24시간 작동하며, DAO는 지능형 거버넌스로 진화합니다. 이는 기존 모델의 최적화가 아니라, 경제적 상호작용의 새로운 형태(New Form of Economic Interaction) 입니다.

이 변화는 이미 시작되었습니다. 초기 도입 사례에서는 결제 비용 100배 절감, 지연 시간 50배 개선, 기존에 불가능했던 새로운 비즈니스 모델이 보고되고 있습니다. 이는 단순한 기술 혁신이 아니라, 인터넷의 탄생에 비견되는 플랫폼 전환(Platform Shift) 의 서막입니다.

8.4 향후 방향 The Path Forward

Kite의 로드맵은 결제 영역을 넘어 검증 가능한 연산으로 확장되고 있습니다. 영지식 자격 증명은 프라이버시를 보호하며 인증을 가능하게 하고, 검증 가능한 추론은 AI의 의사결정을 증명합니다. 이동 가능한 평판은 보편적 신뢰를 창출하며, 서비스 검색은 에이전트 간 자율 네트워크를 형성합니다. 결국 모든 상호작용은 검증되고, 추적되며, 프로그래머블한 형태로 진화할 것입니다.

그러나 이 모든 기반은 이미 존재합니다. 블록체인은 작동 중이며, 프로토콜은 정의되었고, 표준은 구현되었습니다. 개발자들은 오늘 바로, 수십억 개의 에이전트가 수조 건의 트랜잭션을 실행할 수 있는 인프라 위에서 구축을 시작할 수 있습니다.

8.5 행동 촉구 A Call to Action

에이전트 경제가 기다려온 것은 안전하게 자율적으로 작동할 수 있는 인프라입니다. 이제 그 인프라가 존재합니다. 개발자에게 Kite는 기존 플랫폼에서는 불가능했던 에이전트 네이티브 애플리케이션을 구축할 수 있는 기본 프리미티브를 제공합니다. 기업에게 Kite는 암호학적 컴플라이언스와 한정된 리스크를 보장하는 AI 배포 환경을 제공합니다. 사용자에게 Kite는 단순한 챗봇을 신뢰 가능한 금융 행위자로 전환시킵니다.

인간 중심에서 에이전트 네이티브로의 전환은 다음 세대의 대규모 플랫폼 전환을 의미합니다. 모바일 컴퓨팅이 사용자 인터랙션의 모든 전제를 재정의했듯, 에이전트 경제는 아이덴티티, 결제, 그리고 신뢰에 대한 모든 전제를 다시 써야 합니다. 이 변화를 위한 인프라는 더 이상 이론이 아닙니다. 이미 작동 중이며, 확장 가능하고, 앞으로의 10년을 정의할 애플리케이션을 위한 준비가 끝났습니다. 에이전트 경제는 Kite와 함께 시작됩니다.

미래는 단순히 자동화되는 것이 아닙니다. 그것은 자율적입니다. 그리고 그 시작은 지금, **Kite**와 함께입니다.

참고 문헌

[1] Anthropic. (2024). Model Context Protocol (MCP). Retrieved from https://www.anthropic.com/news/model-context-protocol
[2] BIP-32. (2012). Hierarchical Deterministic Wallets. Bitcoin Improvement Proposal. Retrieved from https://github.com/bitcoin/bips/blob/master/bip-0032.mediawiki
[3] EIP-712. (2017). Ethereum typed structured data hashing and signing. Ethereum Improvement Proposal. Retrieved from https://eips.ethereum.org/EIPS/eip-712
[4] EIP-1559. (2021). Fee market change for ETH 1.0 chain. Ethereum Improvement Proposal. Retrieved from https://eips.ethereum.org/EIPS/eip-1559

[5] EIP-4337. (2023). Account Abstraction Using Alt Mempool. Ethereum Improvement Proposal. Retrieved from https://eips.ethereum.org/EIPS/eip-4337
[6] ERC-3009. (2020). Transfer With Authorization. Ethereum Request for Comments. Retrieved from https://eips.ethereum.org/EIPS/eip-3009
[7] ERC-4337. (2023). Account Abstraction via Entry Point Contract specification. Ethereum Request for Comments. Retrieved from https://www.erc4337.io/
[8] European Union. (2024). EU AI Act. Regulation on Artificial Intelligence. Retrieved from https://digital-strategy.ec.europa.eu/en/policies/regulatory-framework-ai
[9] Google. (2025). Agent-to-Agent (A2A) Protocol Specification. Google AI Developer Documentation.
[10] Google. (2024). Agent Payments Protocol (AP2) Specification. Google AI Developer Documentation.
[11] IETF. (2015). RFC 7515: JSON Web Signature (JWS). Internet Engineering Task Force. Retrieved from https://datatracker.ietf.org/doc/html/rfc7515
[12] IETF. (2015). RFC 7519: JSON Web Token (JWT). Internet Engineering Task Force. Retrieved from https://datatracker.ietf.org/doc/html/rfc7519
[13] McKinsey Global Institute. (2023). The economic potential of generative AI: The next productivity frontier. McKinsey & Company.
[14] OAuth 2.1. (2023). The OAuth 2.1 Authorization Framework. Draft IETF Specification. Retrieved from https://datatracker.ietf.org/doc/draft-ietf-oauth-v2-1/
[15] Paradigm. (2025). Tempo: A Payments-First Blockchain Architecture. Paradigm Research.
[16] Protocol Buffers. (2024). Google's language-neutral, platform-neutral extensible mechanism for serializing structured data. Retrieved from https://protobuf.dev/
[17] California State Legislature. (2018). SB 1001: Bots Disclosure Act. California Legislative Information.
[18] W3C. (2022). Decentralized Identifiers (DIDs) v1.0. World Wide Web Consortium. Retrieved from https://www.w3.org/TR/did-core/
[19] W3C. (2022). Verifiable Credentials Data Model v1.1. World Wide Web Consortium. Retrieved from https://www.w3.org/TR/vc-data-model/
[20] Buterin, V. (2014). Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform. Ethereum White Paper.
[21] ECDSA. (2013). Elliptic Curve Digital Signature Algorithm. ANSI X9.62-2005. American National Standards Institute.
[22] EdDSA. (2017). Edwards-curve Digital Signature Algorithm. RFC 8032. Internet Engineering Task Force.
[23] ENS. (2017). Ethereum Name Service Documentation. Retrieved from https://docs.ens.domains/
[24] FATF. (2021). Updated Guidance for a Risk-Based Approach to Virtual Assets and Virtual Asset Service Providers. Financial Action Task Force.
[25] ISO 20022. (2022). Universal financial industry message scheme. International Organization for Standardization.
[26] Lightning Network. (2016). The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments. Poon, J., & Dryja, T.

[27] MCC Codes. (2024). Merchant Category Codes. ISO 18245. International Organization for Standardization.
[28] Merkle, R. (1987). A Digital Signature Based on a Conventional Encryption Function. Advances in Cryptology — CRYPTO '87.
[29] Raiden Network. (2017). Fast, cheap, scalable token transfers for Ethereum. Retrieved from https://raiden.network/
[30] State Channels. (2018). Generalized State Channels on Ethereum. Miller, A., Bentov, I., Kumaresan, R., & McCorry, P.
[31] X402. (2024). Agent Payment Standard Specification. Industry Consortium Draft.
[32] Zero Knowledge Proofs. (1985). The Knowledge Complexity of Interactive Proof Systems. Goldwasser, S., Micali, S., & Rackoff, C.
[33] BLS Signatures. (2001). Short Signatures from the Weil Pairing. Boneh, D., Lynn, B., & Shacham, H.
[34] Travel Rule. (2019). IVMS101 - interVASP Messaging Standard. Joint Working Group on interVASP Messaging Standards.
[35] EUF-CMA Security. (1988). A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks. Goldwasser, S., Micali, S., & Rivest, R.

Kite AI 토크노믹스

Kite AI 미션

Kite AI는 에이전틱 결제를 위한 최초의 블록체인을 구축하고 있습니다. 이 플랫폼은 자율형 AI 에이전트가 검증 가능한 아이덴티티, 프로그래머블 거버넌스, 그리고 매끄러운 결제 인프라를 기반으로 작동할 수 있도록 설계되었습니다. Kite의 목적 지향형 레이어 1 블록체인과 에이전트

패스포트(Agent Passport) 시스템은 AI 에이전트가 1급 경제 주체로 기능할 수 있도록 지원하며, 조합 가능한 상호작용을 통해 새로운 집합적 능력을 창출합니다. Kite AI는 Databricks, Uber, UC Berkeley 출신의 AI 및 데이터 인프라 전문가들에 의해 설립되었으며, PayPal, General Catalyst, Coinbase Ventures, 그리고 주요 블록체인 재단들로부터 총 3,300만 달러를 유치했습니다.

Kite AI 네트워크 개요

Kite AI 블록체인은 지분증명(Proof-of-Stake, PoS) 기반의 EVM 호환 레이어 1 체인으로, 자율형 에이전트 간 상호운용에 필요한 저비용·실시간 결제 및 조정(Coordinator) 레이어의 역할을 수행합니다. L1과 함께 다양한 모듈들이 존재하며, 각 모듈은 데이터, 모델, 에이전트 등 검증된 AI 서비스를 사용자에게 제공합니다. 이러한 모듈들은 특정 산업 또는 용도에 최적화된 환경을 제공하는 준독립형 생태계로, 결제와 정산을 위해 L1과 상호작용합니다. L1과 모듈은 상호 긴밀히 연결된 하나의 통합 생태계를 구성하며, 사용자는 모듈 소유자(Module Owner), 검증자(Validator), 위임자(Delegator) 중 하나의 역할을 수행할 수 있습니다. 네이티브 토큰인 KITE는 인센티브, 스테이킹, 그리고 거버넌스를 구동하는 핵심 자산으로 작동합니다.

KITE 토큰 유틸리티

KITE 토큰은 2단계(Phase)에 걸쳐 기능이 확장됩니다. 1단계에서는 토큰 생성(TGE) 시점부터 초기 참여자가 네트워크에 즉시 참여할 수 있도록 핵심 기능이 제공되며, 2단계에서는 메인넷 출시에 맞춰 고도화된 기능이 추가됩니다.

1단계**(Phase 1)**

모듈 유동성 요건**(Module Liquidity Requirements):** 자체 토큰을 발행한 모듈 소유자는 모듈을 활성화하려면 모듈 토큰과 KITE가 페어링된 영구적 유동성 풀(Permanent Liquidity Pool) 에 KITE를 예치해야 합니다. 이 요건은 모듈의 규모와 사용량에 따라 비례적으로 증가하며, 깊은 유동성을 형성하는 동시에 유통 중인 KITE의 일부를 잠금 상태로 전환합니다. 모듈이 활성 상태로 유지되는 동안 이러한 유동성 포지션은 인출할 수 없으므로, 가장 높은 가치를 창출하는 참여자들의 장기적 토큰 커밋먼트(Commitment)가 보장됩니다.

생태계 접근 및 참여 자격**(Ecosystem Access & Eligibility):** 빌더 및 AI 서비스 제공자는 Kite 생태계에 참여하려면 일정량의 KITE를 보유해야 합니다. 이를 통해 KITE는 에이전틱 결제 생태계에 참여하기 위한 접근 토큰(Access Token) 으로서 즉각적인 효용성을 가지며 비즈니스 및 개발자들의 이해관계를 Kite 네트워크 전체의 성장 방향과 정렬시킵니다.

생태계 인센티브**(Ecosystem Incentives):** KITE 공급량의 일부는 Kite 생태계에 실질적인 가치를 창출하는 사용자 및 비즈니스 참여자에게 인센티브 보상 형태로 분배됩니다.

2단계**(Phase 2)**

AI 서비스 수수료 **(AI Service Commissions):** 프로토콜은 각 AI 서비스 트랜잭션에서 발생한 수수료 일부를 징수해, 이를 시장에서 KITE로 스왑한 뒤 모듈과 L1에 분배합니다. 이 구조를

통해 서비스 운영자는 자신이 선호하는 통화(예: 스테이블코인)로 수익을 받을 수 있으며, 모듈과 L1은 네이티브 토큰(KITE)으로 보상을 받아 네트워크 내 지분(Stake) 과 영향력(Influence) 을 확장할 수 있습니다. 결과적으로 프로토콜의 수익은 스테이블코인 매출에서 KITE로 전환되며, 이 과정에서 AI 서비스의 실제 사용량과 매출에 직접 연동된 지속적인 매수 압력(buy pressure) 이 형성됩니다. 이에 따라 KITE의 가치는 네트워크 채택과 함께 안정적으로 성장하도록 설계되어 있습니다.

스테이킹 **(Staking):** KITE 스테이킹은 네트워크 보안을 강화하고, 사용자는 보상과 교환으로 서비스를 수행할 수 있는 자격을 얻게 됩니다.

- 모듈 소유자(Module Owner) 는 Kite 네트워크 상에서 모듈을 개발·운영·관리합니다.
- 검증자(Validator) 는 지분증명 합의(Proof-of-Stake)에 참여해 네트워크를 검증합니다.
- 위임자(Delegator) 는 자신이 가치 있다고 판단한 모듈을 지원하기 위해 토큰을 예치합니다.

거버넌스 **(Governance):** 토큰 보유자는 프로토콜 업그레이드, 인센티브 구조, 모듈 성과 요건 등에 대해 투표하며 이해관계자들의 이익이 네트워크의 장기적 지속가능성과 일치하도록 합니다.

모듈**,** 검증자**,** 위임자**(Modules, Validators, and Delegators)**

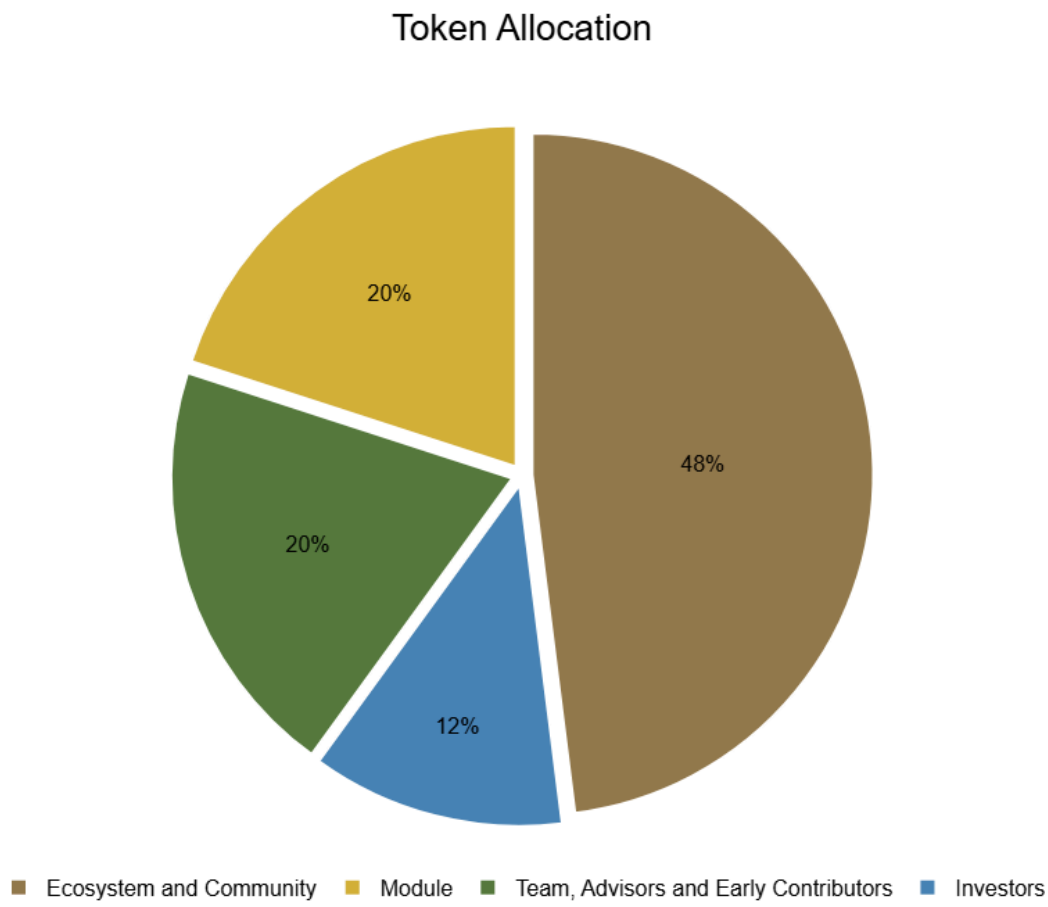
Kite 검증자는 토큰을 스테이킹하고 합의(Consensus)에 참여함으로써 네트워크를 보호합니다. 각 검증자는 특정 모듈을 선택하여 스테이킹하며, 해당 모듈의 성과와 인센티브가 정렬되도록 합니다. 검증자의 주요 역할은 네트워크 보안과 합의 안정성을 유지하고, 거버넌스에 참여해 투표를 수행하며, 커뮤니티 협업과 생태계 성장을 촉진하는 것입니다.

Kite 위임자는 토큰을 스테이킹하여 네트워크를 보호합니다. 각 위임자는 반드시 특정 모듈에 스테이킹해야 하며, 그 모듈의 성과에 따라 인센티브를 공유합니다. 위임자는 생태계 거버넌스에 참여하고 커뮤니티 활동에 기여할 것으로 기대됩니다.

모듈, 검증자, 위임자는 장기 보유를 유도하도록 설계된 지속적 보상 메커니즘(Continuous Reward System)을 통해 토큰 보상을 받습니다. 참여자는 일정 기간 동안 '저금통(piggy bank)' 형태로 KITE 보상을 누적하며, 언제든 누적된 토큰을 청구하거나 매도할 수 있습니다. 그러나 이를 청구하거나 매도할 경우 해당 주소의 향후 모든 토큰 발행은 영구적으로 중단됩니다. 이 메커니즘은 즉각적 유동성과 장기적 가치 축적 간의 균형을 유도함으로써, 단기 보상 수혜자를 네트워크의 지속 가능한 이해관계자(Long-term Aligned Stakeholder)로 전환합니다.

초기 **KITE** 토큰 분배**(Initial KITE Allocation)**

KITE의 총 공급량은 100억 개(10,000,000,000 KITE)로 상한이 설정되어 있으며 다음과 같이 분배됩니다.



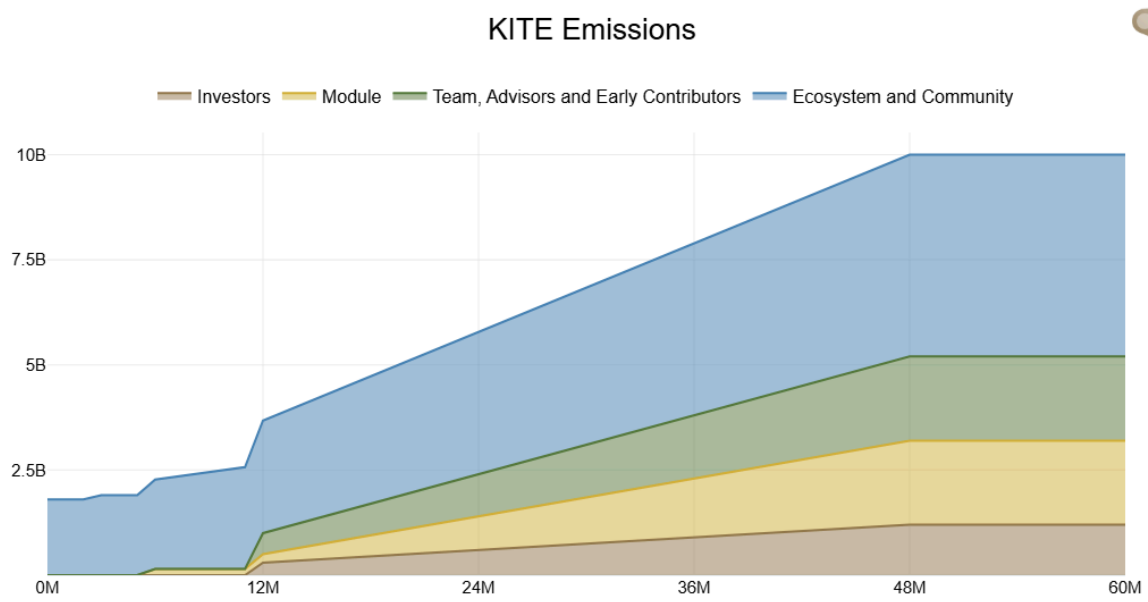
생태계 및 커뮤니티**(Ecosystem and Community, 48%):** 생태계 및 커뮤니에 배정된 토큰은 사용자 채택을 가속화하고, 빌더와 개발자의 참여를 촉진하며, 생태계 유동성을 강화하는 데 사용됩니다. 이 토큰은 커뮤니티 에어드롭, 유동성 프로그램, 성장 인센티브 등을 통해 참여를 확대하고, 의미 있는 기여를 보상함으로써, Kite가 출시 단계를 넘어 실질적 효용 기반 생태계로 발전할 수 있도록 지원합니다.

투자자**(Investors, 12%):** 투자자에게 배정된 토큰은 구조화된 베스팅(vesting) 일정에 따라 분배되며, 투자자의 이해관계를 네트워크의 장기적 성장과 일치시키는 것을 목표로 합니다. 이를 통해 초기 재정적 후원자들이 생태계 확장 단계에서도 지속적으로 네트워크에 헌신하도록 유도합니다.

모듈**(Modules, 20%):** 모듈에 배정된 토큰은 고품질 AI 서비스 개발을 장려하고, 사용자와 Kite 생태계 간 원활한 상호작용을 지원하는 인프라 확장에 사용됩니다. 해당 토큰은 개발자 보조금(Developer Grants), 성과 기반 보상, 그리고 네트워크의 지능과 접근성을 동시에 강화하는 서비스 구축에 활용됩니다.

팀, 어드바이저 및 초기 기여자**(Team, Advisors, and Early Contributors, 20%):** 팀과 어드바이저, 초기 기여자에게 배정된 토큰은 Kite 빌더들의 장기적 인센티브를 네트워크의

지속 가능한 성공과 일치시키는 역할을 합니다. 해당 토큰은 초기 기여자, 개발자, 전략적 어드바이저에게 분배되며, 생태계 성숙 단계에서 안정성과 책임성을 확보할 수 있도록 다년간의 베스팅 일정에 따라 지급됩니다.



Kite 네트워크의 가치 포착 구조**(The Kite Network Value Capture)**

매출 기반 네트워크 성장**(Revenue-Driven Network Growth):** Kite AI는 토큰 가치를 네트워크의 실제 매출과 사용량에 직접 연계하는 다양한 메커니즘을 도입하고 있습니다. AI 서비스 트랜잭션에서 발생하는 수수료의 일정 비율은 모듈과 네트워크의 커미션 형태로 징수되며, 이를 통해 네트워크 내 모든 거래가 토큰 가치 상승으로 직접 연결되도록 설계되어 있습니다. 모듈이 성장하고 더 많은 매출을 창출할수록 추가적인 KITE가 해당 모듈의 유동성 풀에 잠기며, 네트워크 전반의 유동성과 토큰 수요가 함께 강화됩니다.

비인플레이션형 경제 구조**(Non-Inflationary Economics):** 기존의 지분증명(Proof-of-Stake) 네트워크들이 지속적인 토큰 인플레이션에 의존하는 것과 달리, KITE는 발행 기반 보상(emission-based rewards)에서 신속히 전환하여 프로토콜 매출에 기반한 지속 가능한 모델을 구축합니다. 초기에는 전용 보상 풀을 통해 발행된 토큰이 네트워크 참여를 부트스트랩하지만, 시스템은 실제 AI 서비스 사용에 기반한 매출형 보상 구조로 점진적으로 이행하도록 설계되어 있습니다. 이를 통해 토큰 보유자는 인플레이션으로 인한 희석 위험에서 벗어나고, 보상은 생태계 내에서 창출된 실질적 가치에 의해 직접 지급됩니다.

이 같은 메커니즘은 실제 AI 수요가 토큰 가치를 직접적으로 견인하는 선순환 구조를 형성합니다. 서비스 사용이 매출을 창출하고, 성공적인 모듈이 더 많은 유동성을 예치하며, 높은 가치를 창출하는 참여자들이 네트워크 성장에 지속적으로 기여하도록 인센티브를

받습니다. 이러한 구조를 통해 KITE는 네트워크의 효용성과 채택률에 본질적으로 연동된 토큰 경제 시스템을 구축하고, Kite AI 네트워크의 성장과 함께 그 가치가 안정적으로 확장되도록 설계되어 있습니다.