저자: Joseph Chow

상태: 초안

유형: Meta

생성날짜: 2015년 11월 17일

**요약**

이 문서는 BIP 123에 의해서 채택된, EIP의 분류 체계에 대해서 설명합니다.

EIP는 더 복잡한 정보의 상호 운용처리(정보와 관련하여 기기간 그리고 이용자 관계에서 정보를 원활하게 또한 충분히 교환, 처리할 수 있는 것)을 포함한, 번호가 낮은 레이어 시스템으로 분류됩니다.

이 설명은 레이어를 정의하고, 어떤 레이어가 특정한 표준 EIP에 속하는지 결정하기 위한 기준을 설정합니다.

동기

이더리움은 몇몇의 다른 기준들을 포함하는 시스템이다. 어떤 기준은 정보의 상호 윤영처리를 위한 필수적인 요소인 반면, 어떤 기준은 선택적으로 보완을 지원해 수 있는 옵션으로 여겨질 수 있다.

정보의 상호 운용 처리 요구를 더 잘 반영한 EIP 프로세스를 갖기 위해서는 EIP를 분류하는 것이 필수적이다. 낮은 레이어는 표준을 받아들이고 적용하는데 상당히 큰 어려움을 갖고 있다.

**명시**

표준 EIP는 4가지 레이어로 나눌 수 있다.

1. 컨센서스
2. 네트워킹
3. API/RPC
4. Applications
5. **Consensus Layer**

컨센서스 레이어는 암호학적 약속의 방식을 정의하고 있다. 이것의 목적은 누구든 특정한 상태와 기록이 타당한지 지역적으로 평가할 수 있고, 합의의 보장을 제공하고, 궁극적인 접합점을 보장하는 것이다.

컨센서스 레이어는 메시지가 어떻게 네트워크에 전파되는지를 고려하는 것이 아니다.

컨센서스 레이어에 대한 불일치는 다른 노드들이 각각 호환되지 않는 다른 기록을 받아드리게 되는 네트워크의 분할이나 포크를 야기할 수 있다. 우리는 더 나아가 이러한 레이어의 변화(앞서 언급한 네트워크의 분할)을 소프트 포크와 하드 포크로 세분할 수 있다.

**소프트포크**

소프트 포크에서, 예전의 규칙에서는 유효한 어떤 구조는 더 이상 새로운 구조에서는 유효하지 않다. 그리고 예전의 룰에서는 타당하지 않던 구조가 지속적으로 새로운 룰에서도 타당하지 않다. [길게 늘어진 체인을 상상해보자면, ㅁ-ㅁ-ㅁ-ㅁㅅ-ㅁㅅ-ㅁㅅ 이런 식으로 소프트 포크가 진행되었다고 쉽게 이해할 수 있다. ‘ㅁ’을 예전의 룰, ‘ㅁㅅ’을 새로운 룰이라 가정하자. 이때, ‘ㅁㅈ’은 예전 구조에서 새로 업그레이딩 된 규칙이라고 하자. 예전의 규칙에서 유효했던 구조인 ’ㅁ’은 더 이상 새로운 구조에는 유효하지 않다(새로운 룰은 ㅁ을 업그레이딩 한 상태이므로) 또한, 참인 한 전제의 대우는 언제나 참이므로, ‘예전의 룰에서 타당하지 않던 구조는 새로운 룰에서도 타당하지 않다’는 참인 전제의 대우명제인 ‘새로운 룰에서 타당한 구조는 예전의 룰에도 타당한 하다’를 살펴보자. 새로운 업그레이딩 된 룰’ㅁㅅ’에서 타당하다면, 예전의 룰’ㅁ’에서도 타당하게 되는 것이다. 즉, 새로운 룰은 예전의 룰을 포함하면서 더욱 업그레이딩 된 상태인 것이다.]

--

**하드 포크**

하드 포크에서. 예전의 룰에서는 타당하지 않던 구조가 새로운 구조에서는 타당하다. (이번에도 마찬가지로 길게 늘어진 체인을 예로 들어서 설명하자면, 소프트 포크와 다르게 하드 포크는 체인이 두 갈래로 쪼개진다. ㅁ-ㅁ-ㅁ-**ㅁ-ㅁ** 그리고 ㅁ-ㅁ-ㅁ-**ㅅ-ㅅ** 기존을 유지하는 체인과 새로이 바뀌게 된 체인이 쪼개져서 존재하게 된다. ‘ㅁ-ㅁ-ㅁ’ 이후로 ‘ㅁ-ㅁ’이라는 체인과 ‘ㅅ-ㅅ’이라는 체인이 두 갈래로 존재하게 되는 것이다. 따라서 예전 룰에 타당하지 않던 ‘ㅅ’이라는 구조는 새로운 구조에는 타당하게 된다)

1. **Networking Layer**

네크워킹 레이어는 이더리움 와이어 프로토콜(eth)과 라이트 이더리움 서브프로토콜(les)을 명시합니다. RLPx는 [https://github.com/ethereum/devp2p devp2p 저장소에서 배제되고 추적된다

오직 기본 노드 상호 운용을 위해 서브프로토콜의 서브세트만 필요로 한다. 노드들은 추가적인 확장 기능을 지원할 수 있다.

이것은 기존에 존재하는 프로토콜의 호환성을 해치지 않으면서, 오히려 점차적으로 이전의 프로토콜을 대체하면서, 새로운 서브프로토콜을 더할 수 있다. 이러한 방식으로 전체 네트워크는 서비스 중단의 심각한 위험 없이 전체 네트워크를 업그레이드할 수 있다.

1. API/RPC Layer

API/RPC 레이어는 응용프로그램이 접근할 수 있는 더 상위 수준의 콜(call)을 명시할 수 있다.

이 레이어에는 기본 네트워크 상호 운용성을 해치지 않으면서 경쟁 표준을 허용할 여지가 있습니다.

1. Applications Layer

어플리케이션 레이어는 다른 어플리케이션이 비슷한 기능을 지원하고 데이터를 공유할 수 있게 해주는 상위수준의 구조, abstraction 그리고 규칙들을 명시한다