한 번에 끝내는 블록체인, dApp 개발의 모든 것

Chapter 4

Lottery 컨트랙트 v2 개발

Chapter 4

Lottery 컨트랙트 v2 개발

Chainlink의 VRF란

Chainlink VRF란

- 스마트 컨트랙트가 더이상 보안 및 활용성을 타협하지 않으면서 랜덤 값을 얻을 수 있도록 해주는 입증 가능하게 공정하고, 예측이 불가능한 랜덤 숫자 생성기
- 각 요청마다, Chainlink VRF는 하나 이상의 랜덤 값과 각 랜덤 값이 어떻게 결정됐는지에 대한 암호학적 증명을 생성함
- 증명은 요청한 스마트 컨트랙트가 랜덤 값을 사용할 수 있게 되기 전에 온체인에서 검증됨

VRF가 필요한 이유

- VRF: Verifiable Random Function
- 랜덤 값이 실제로 안전하게 생성됐는지에 대한 증명이 없다면, 이를 사용하는 사람들은 맹목적으로 이 값이 안전한 랜덤 값이라고 믿고 사용하는 수밖에 없음
- 실제로도 누구의 개입 없이 안전하고 랜덤하게 생성된 값이라는 증명이 가능해야만 trust-minimized 서비스 구축 가능

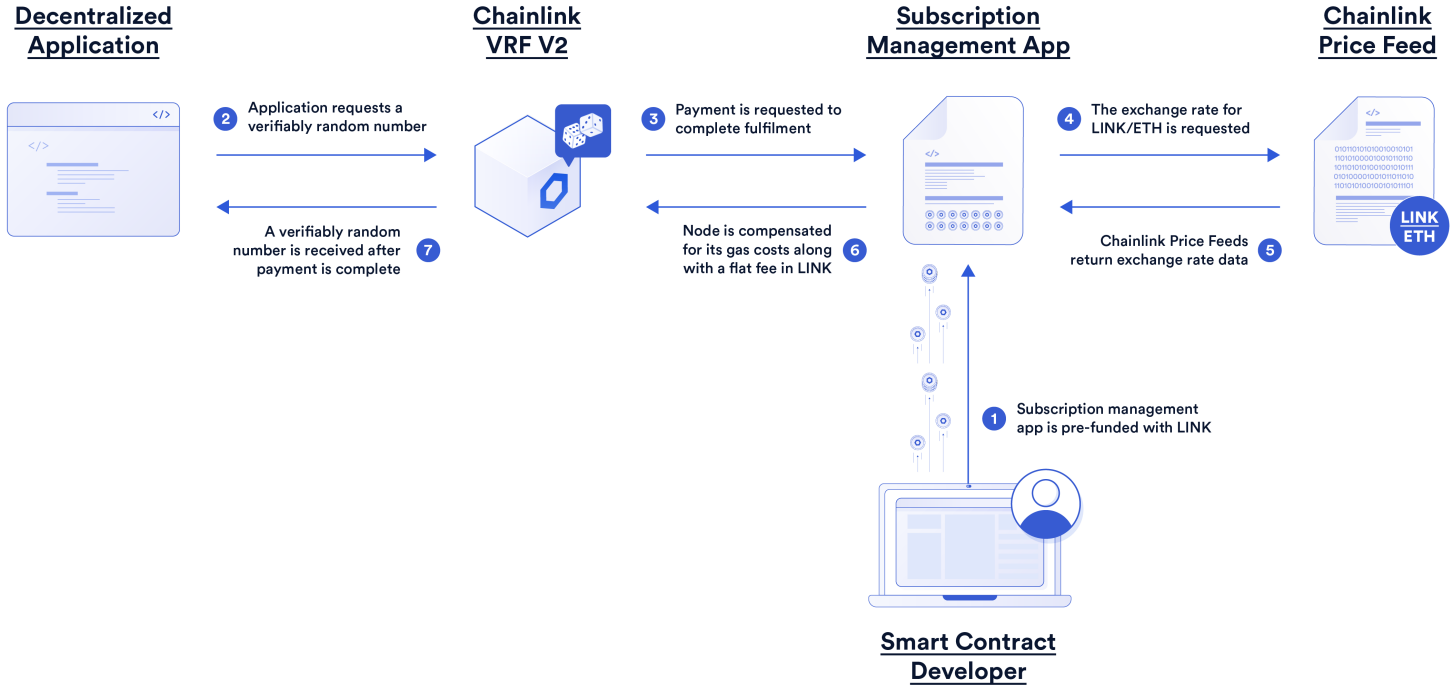
Chainlink VRF v2 특징

---

1. 개발자가 Chainlink의 subscription management app에 미래에 사용할 LINK 토큰을 미리 충분히 예치해 둘 수 있고, 여러 컨트랙트를 등록해 놓을 수 있음 → v1에서 각 컨트랙트마다 사용하기 전에 충분한 LINK 토큰을 전송해놓은 후 사용했어야하는 불편함 개선
2. callbackGasLimit을 둠으로써 Chainlink VRF로부터 랜덤 값을 callback을 통해 받을 때 더 복잡한 로직 수행 가능
3. block confirmation 파라미터 조정 가능 → 개발자가 구현 중인 서비스의 필요에 따라, 안전성을 지키면서 (block re-organazation으로부터 보호) 퍼포먼스(랜덤 값 요청 후 랜덤 값 받을 때까지 기다리는 시간)를 핸들링 가능해짐
4. 한 번의 요청을 통해 여러개의 랜덤 값을 받도록 할 수 있음

Chainlink VRF (v2) 동작 방식

1. 스마트 컨트랙트 개발자가 Chainlink의 subscription management app에 LINK 토큰 예치 및 랜덤 값 요청할 컨트랙트 등록
2. 컨트랙트에서 Chainlink VRF로 랜덤 값 요청
3. Chainlink VRF는 해당 컨트랙트에 대해 subscription management app에 예치되어있는 LINK 토큰을 사용하여 랜덤 값 생성 및 검증
4. 검증된 랜덤 값을 요청한 컨트랙트로 전송



Chainlink VRF Consumer

- Chainlink VRF에 랜덤 값을 요청하는 컨트랙트에서 필요한 구조
- requestRandomWords(): Chainlink VRF로 랜덤 값을 요청하는 함수
- fulfillRandomWords(): Chainlink VRF에서 검증 완료된 랜덤 값 생성 후 callback으로 호출하는 함수

```
// Assumes the subscription is funded sufficiently.
function requestRandomWords() external onlyOwner {
  // Will revert if subscription is not set and funded.
  s_requestId = COORDINATOR.requestRandomWords(
    keyHash,
    s_subscriptionId,
    requestConfirmations,
    callbackGasLimit,
    numWords
  );
}

function fulfillRandomWords(    undefined gas
  uint256, /* requestId */
  uint256[] memory randomWords
) internal override {
  s_randomWords = randomWords;
}
```