1. 12페이지 전자 서명관련해서: 블록체인의 서명 대상은 “Transaction”이며, 트랜잭션 자체를 암호화하지는 않으며, Transaction의 Hash값을 Private Key를 가지고 서명(암호화) 시켜서 보내게 됩니다. 받은 측에서는 “Transaction”과 그 암호화된 서명 값을 알고, 보낸 사람의 주소에서 Public Key를 복구(Recover)해서 해시 값을 풀어(복호화)내게 되고, 이렇게 얻은 값으로 전달받은 “Transaction”의 해시 값을 구해서 비교해 제대로 서명이 되었는지를 확인하게 됩니다.
2. 15페이지 원장을 갱신: 원장이 갱신되는 것이 아니라 계정 정보가 바뀝니다. 즉, 관리하는 자산의 상태가 트랜잭션에 따라 변경됩니다. 원장은 기록이며 블록에 저장됩니다.
3. 15페이지 계정관리: 블록체인 계정(주소)는 보통 블록체인과 무관하게 만들어집니다. 그래서 EOA(이더리움의 경우: Externally Owned Address)라고 부릅니다. 따라서, 여기서 말하는 계정은 특정 블록체인을 플랫폼처럼 서비스를 제공할 때 해당 서비스에 접속하기 위한 계정입니다. 즉, 블록체인 자체는 계정 정보를 저장하지 않습니다.
4. 16페이지 블록체인 외부 시스템 계정 접속: 이 내용은 private blockchain에만 해당하는 사항입니다. Public은 아닙니다.
5. 16페이지 Transaction의 그림이 조금 이상하네요, Ethereum의 경우 from, to, value, data, 전자서명 관련 VRS값 정도 있습니다. 확인하세요. 일반적인 블록체인 트랜잭션을 이야기하시려면, 충분할 것 같지만 Header 정보는 굳이 있어야 할지 모르겠네요.
6. 17페이지 그림: 비트코인 관련 정보라고 보입니다. 아래 블록 검증 데이터로 표시된 부분은 제거하시는 게 일반적인 블록체인 구조라고 보입니다.
7. 21페이지 표: PBFT에 Cosmos정도(Tendermint) 활용 사례로 넣어도 될 것 같네요.
8. 21페이지 표: Ethereum PoS의 경우는 Enterprise Ethereum보다 더 빠릅니다. 실제로 Besu는 250도 안 나올 수 있습니다. 빠름이 아닌 보통 수준일 것 같네요. 그리고, Enterprise Ethereum의 경우 BFT 계열의 합의 알고리즘으로 구현되어 있다고 볼 수 있습니다. PoA는 일종의 대표 이름이고, 내부에 iBFT2, QBFT, Clique등을 사용합니다. Besu는 Java로 구현되어 느릴 수 있습니다.
9. 29페이지 암호화 알고리즘들: Ethereum은 keccak256과 같이 SHA3 계열을 변형해서 사용하고 있습니다. 관련해서 사용되는 실제 알고리즘이 기반한 것을 변형한 것이 있으니 참고하세요.
10. 30페이지: 이것은 계정관리 관련해서 서비스 구축자의 준수 사항이지, 불록체인 자체에 대한 이야기는 아닌 것 같습니다.
11. 39페이지: 이 부분은 약간 Hyperledger Fabric에 대한 내용을 강조한 것 같네요. 다른 블록체인도 고려할 필요가 있어 보입니다. Hyperledger Fabric은 이렇게 하고 있는 것으로 압니다만, 다른 블록체인은 그렇지 않을 수 있습니다.
12. 41페이지: 블록체인은 당연히 그렇게 합니다. 크게 의미가 없는 항목 같습니다. 블록체인 시스템을 정의하는 것이 아니라면 의미 없다고 봅니다.
13. 42페이지: 블록생성자의 전자서명은 검토해 보시기 바랍니다. 악의적인 노드라는 것이 어떤 의미인지 모르겠네요. 예를 들어, Public이라고 한다며 생성을 했다고 해서 연결이 안될 것이고, Private이라고 한다면 악의적인 노드가 합의에 참여 못하게 막을 방법이 없을 것 같습니다. 물론 인증서를 제대로 얻은 노드라고 한다면 당연히 참가할 수 있고, 블록을 만들었다고 해서 다 연결이 되는 것은 아닙니다.
14. 43페이지: 정확한 의미를 잘 모르겠네요.