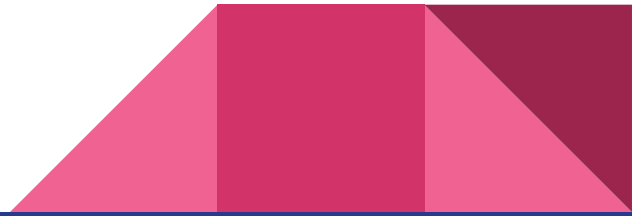
Adamnite

A secure and scalable blockchain development platform

By Archie Chaudhury

Background

- Bitcoin’s invention in 2009 not only revolutionized finance, but also led to the advent of blockchain technology.
- Bitcoin was one of the first applicable BFT distributed ledger systems.
- Bitcoin underlying technology was used for little beyond DeFi.
- Ethereum, developed in 2013, further accelerated the adoption of blockchain technologies in other applications.
- Alternative platforms such as Algorand, Cardano, and Solana have emerged after 2017 to provide faster and more scalable solutions.



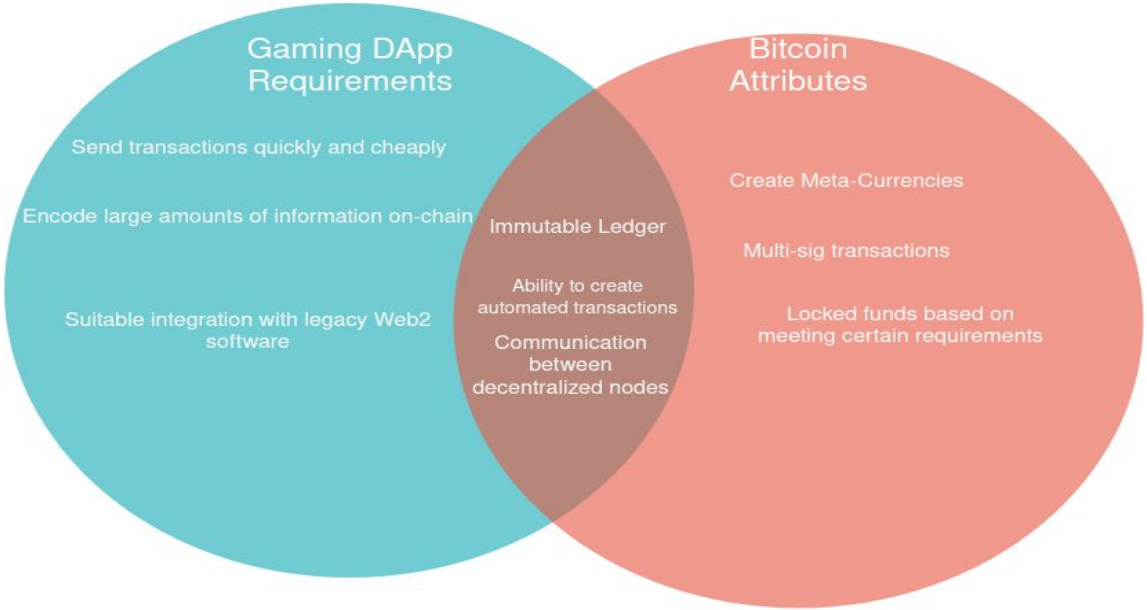
Lack of Scalability

- Bitcoin Script is a stack-based language that allows developers to interact directly with the Bitcoin Blockchain.
- Smart Contracts dictating automatic transfers and actions on the blockchain can be created, although they are difficult.
- Only transactions; programs cannot take into account or mutate on-chain data.
- A difficult syntax and structural limitations impedes the development of more complex applications.

```
scriptPubKey: OP_DUP OP_HASH160 <pubKeyHash> OP_EQUALVERIFY OP_CHECKSIG
scriptSig: <sig> <pubKey>
```

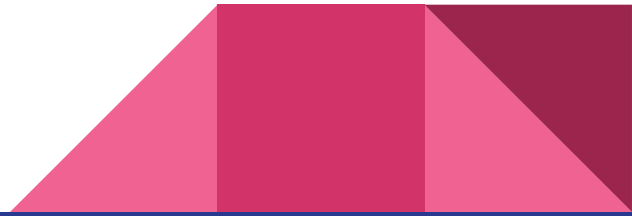
Source: Bitcoin Wiki

Example: Gaming Applications



Lack of Scalability (cont)

- Ethereum: Turing-Complete and allows developers to take the overall state of the blockchain into account.
- Allows developers to build smart contracts that can define decentralized applications, on-chain assets, and more.
- Features multiple high-level languages such as Solidity and Vyper.
- Problem: gas fee optimization, lack of composability, and

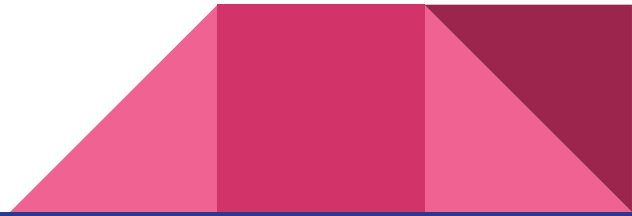


Adamnite

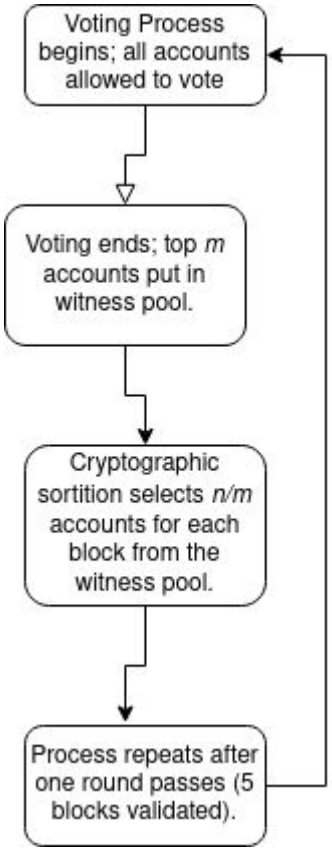
- Secure and scalable blockchain development platform
- Layer-1 Chain
- Modified Delegated Proof of Stake Consensus guarantees efficiency while retaining both decentralization and security.
- In-built security compiler when translating from intermediate form for high-level programming languages; ensures simple mistakes in on-chain applications do not leave room for large exploits.

Adamnite Philosophy

- **Easy to use:** Adamnite's key appeal is its simplicity: its programming language and associated concepts should be as easy to learn as Python, C++, etc.
- **Scalability:** Organizations and businesses should easily implement Adamnite into their current framework without worrying about security, transaction costs, or a high learning curve.
- **Security:** A significant feature of Adamnite's compiler will be an automatic verification engine that checks for both safety and validity.

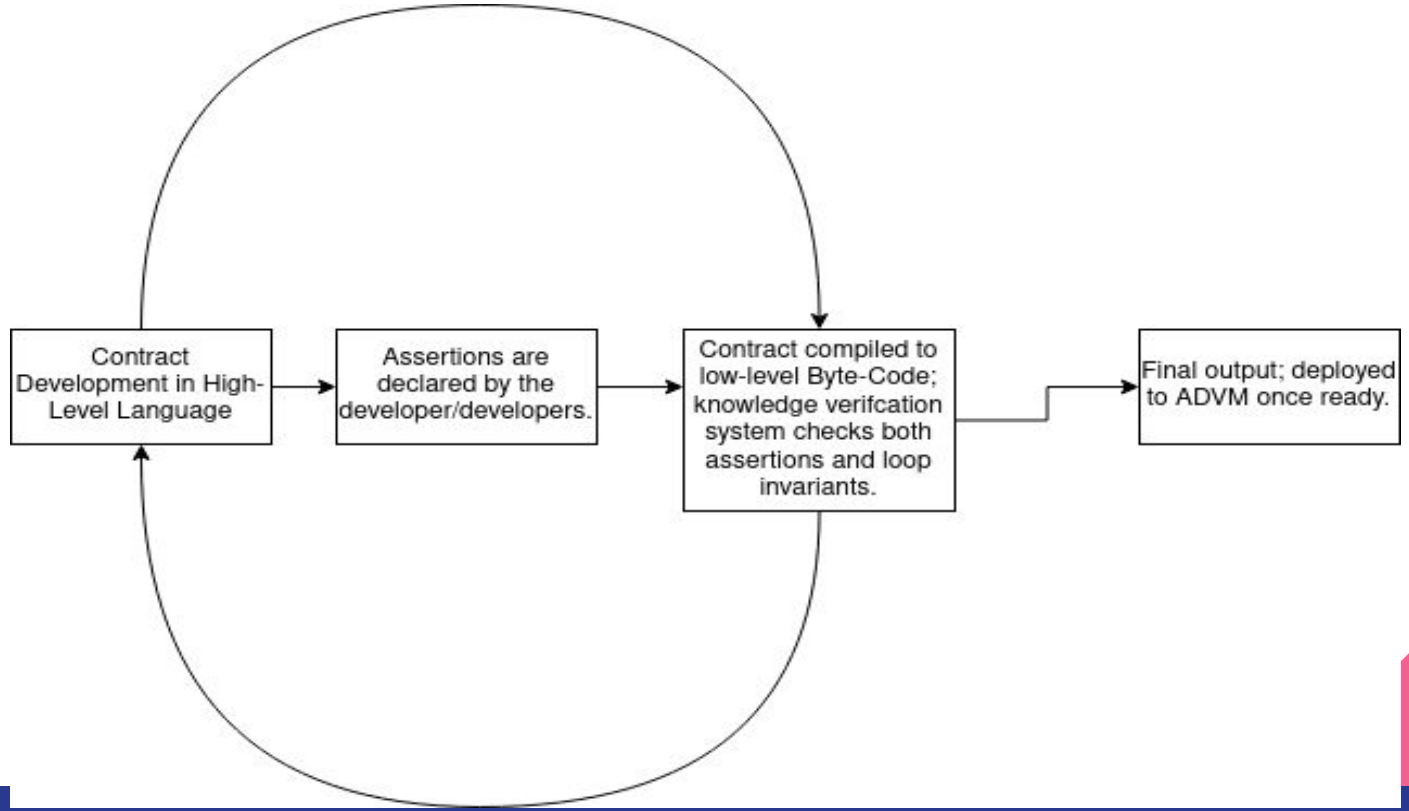


Consensus Mechanism



- Both *n* and *m* depend on the total size of the network.

Knowledge Verification during compilation



Use Cases

- Digital Asset Creation (NFTs, sub currencies, etc)
- Video Streaming/Encoding
- Decentralized Autonomous Organizations (DAOs)
- Gaming
- Social Networks (Web applications, Metaverse)
- Decentralized Finance (Decentralized Exchanges, Lending Protocols)
- More yet to be discovered!