特立独行的CKB

宁志伟

Agenda

分层架构 VS 单一架构

PoW VS PoS

RISC-V VS EVM/WASM

Cell model VS Account model

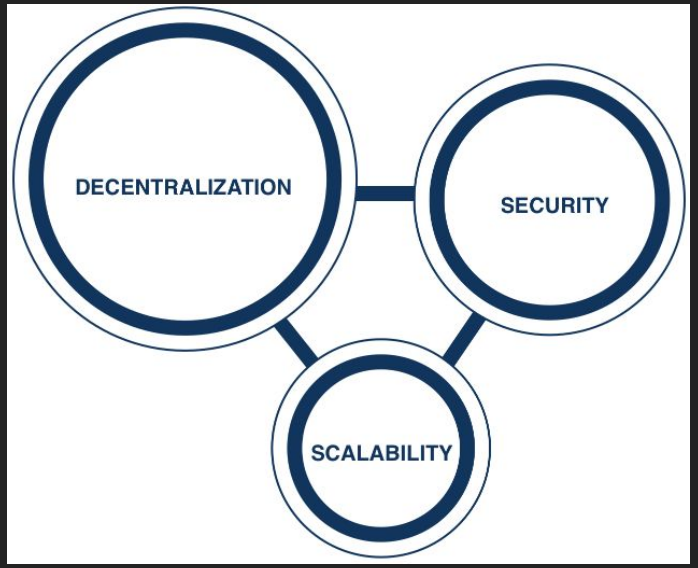
Q&A

区块链的核心价值是提供信任。
区块链是通过全局共识的方式实现信任，而全局共识必然是一个很慢、很贵的东西。

没有共识算法能比没有共识的共识算法更快。
分布式、去中心化系统的共识，不可能做得比中心化的共识更快。
区块链最应该考虑的是如何能够把信任的价值发挥出来。

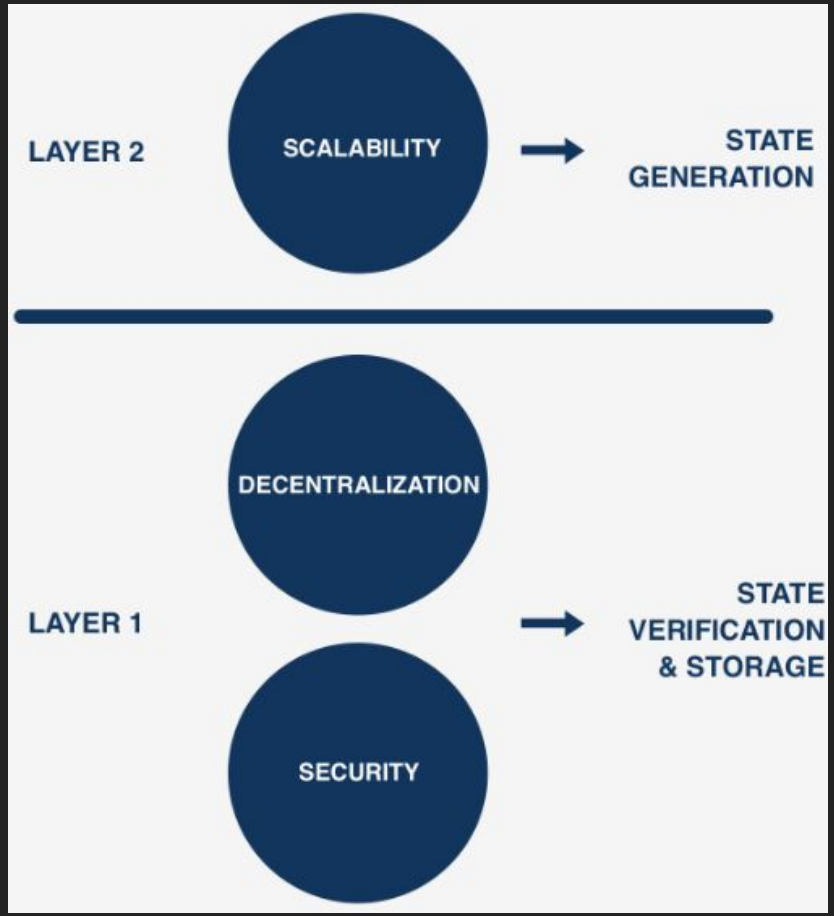
01 分层架构 VS 单一架构

the scalability trilemma



扩容方案都是有代价的:
联合挖矿，大区块，超级节点等方案影响去中心化。
多链，分片等方案降低安全性。

分层架构



Layer1 关注去中心化和安全
Layer2 关注性能

需求也是分层的:
大部分情况下只需要局部的共 识。
越大范围的共识代价越大。
即使中心化系 统也是如此。

不同的角色:状 态和计算。
Layer1 负责状态的验证和存储
Layer2 状态的生成(计算)

02 PoW VS PoS

女巫攻击

共识就是少数服从多数。
最常用的方式就是投票。
女巫攻击其实就是刷票。

PoW和PoS都能解决女巫攻击的问题。

确定性

BFT 有绝对的确定性。
其投票节点是事先确定的。

PoS有一定的确定性。
在指定的时刻，Stake总量是固定的值。

PoW没有最终确定性。
在指定的时刻，算力总量不是一个固定的值。

PoW有动力吸引更多的算力，系统安全性随之上升。新矿工没有门槛。

PoS没有动力去增发Stake。系统的安全性也跟这个没有直接关系。新的验证者门槛较高。

PoW相比之下更开放，安全性更可量化，与Layer2更互补。

ASIC-neutral Proof-of-Work function。

希望算力是分散的。

希望有尽量多的算力。

带宽实际上是区块链吞吐量的最大限制。

NC-Max 有三个主要的创新：
采用两步交易确认来降低孤块率。
动态调整区块间隔和区块奖励来更好的提升带宽利用率。
在难度调整的时候考虑周期中的所有区块，来抵御自私挖矿攻击。

03 RISC-V VS EVM/WASM

VM in BlockChain

apply : tx X chain state → new chain state

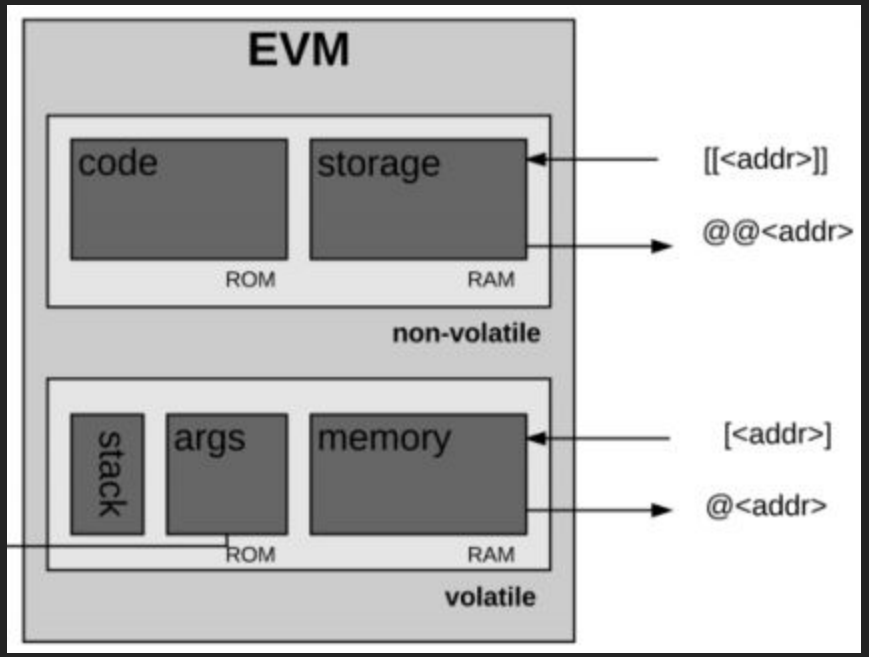
Have to:
Certainty
Halting problem(each instruction has exact price)

Other:
Stable instruction set
Flexible
Toolchain support

Comparison

	RISC-V	WASM	EVM
Instructions	50	172	188
Words size	32/64	32/64	256
Vm Architecture	Register Based	Stack Based	Stack Based
Level	Hardware	Software	Software
toolchain	GCC/LLVM	LLVM	solc
Language	C/Rust	C/Rust etc	Solidity

Memory Model



CKB-VM use 64-bit RISC-V instruction sets
Not bytes code but Linux ELF

No operation stack but 32 registers

4 MB runtime memory(include thread stack/heap/ELF file mapping)

Access chain data by syscalls(only current transaction related information)

04 Cell model VS Account model

Programing model and Problems

Programing model is about State:
state generation (off-chain)
state verification (CKB VM)
state storage (Cell model)

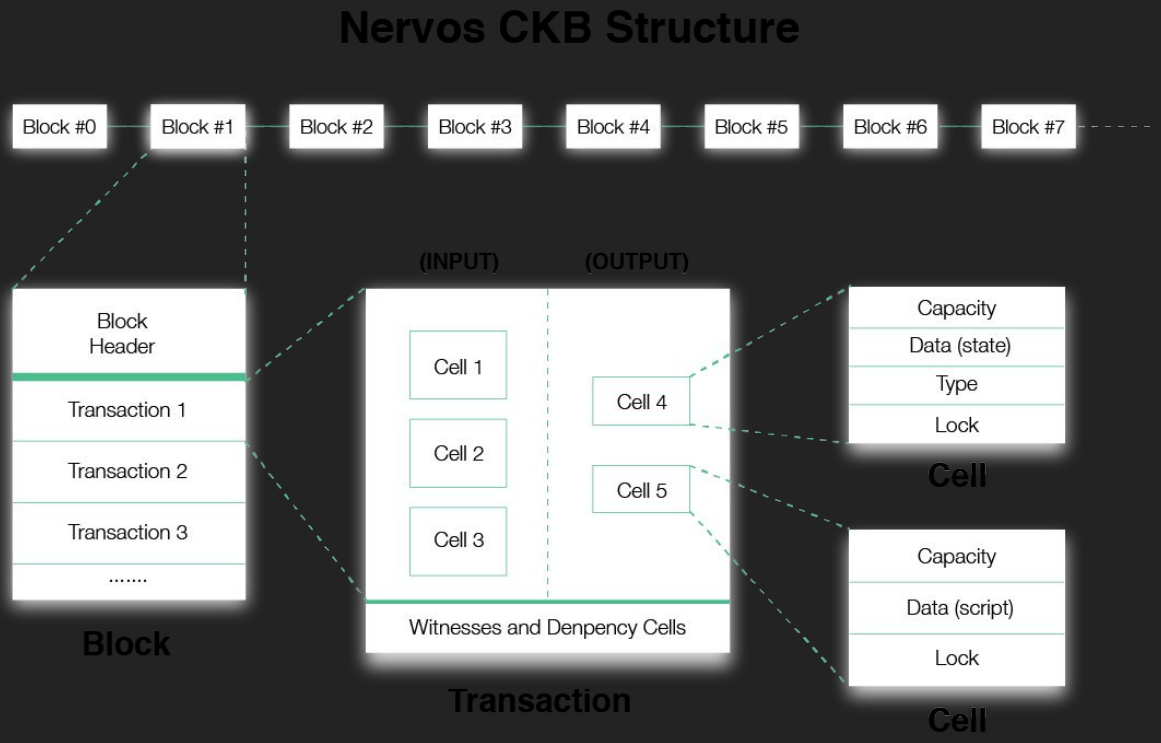
Problems:
We only pay for computation, State storage is free.
The Tragedy of Commons.

Computation and verify should be separated.

Comparison

	Bitcoin	Ethereum	CKB
Instruction Set	Script	EVM	RISC-V
Cryptographic Primitive	Opcode	Precompile	Assembly
Stateful	No	Yes	Yes
State Type	Ledger	General	General
State Model	UTXO	Account	Cell
State Verification	On-chain	On-chain	On-chain
State Generation	Off-chain	On-chain	Off-chain

Cell model



Cell model from generalizing the UTXO model.

Cell model VS Account model

Account model like OOP.
Function in Contract like method in Class.
Storage variable like member of Class.
Transaction like message.
Computation on nodes.

Cell model like Pure FP.
Once a cell used as input, it won't be used any more.

Lock script like Callback.

Computation on client.
Verify on chain(always recompute).

Contract example on ckb

- Most simple script.
- Vote - Map Reduce.
- HTLC. Multi-Signature.

https://github.com/rink1969/ckb-contract-examples

Q&A

Thank You!

www.nervos.org

github.com/nervosnetwork