Factom 2.0

Breaking the Blockchain barrier to Performance

Where is the Industry at?

EOS	9,656
Polkadot	7,200
IOTA	1,250
Hedera (HashGraph)	1,300
Hyperledger	700
Factom	60
Ethereum	44
Bitcoin	7

Limiting TPS evaluations to actual test results, we find that the marketing of various projects is not necessarily reflected in performance measured on Test Nets

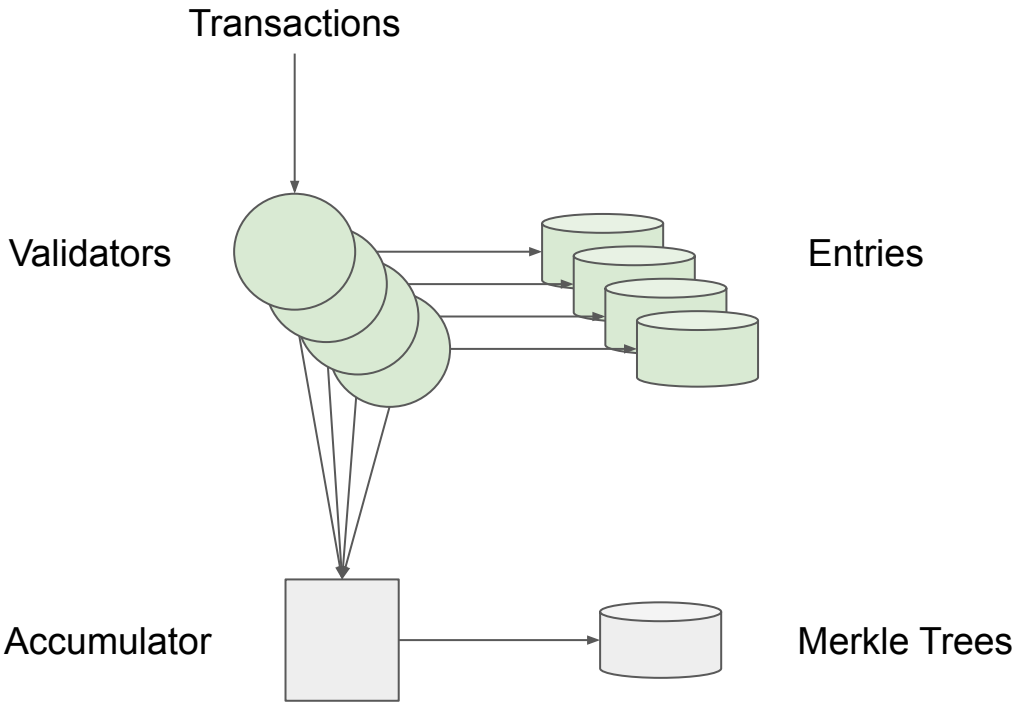
More centralized protocols clearly outperform more decentralized protocols

Creating a hybrid public/private blockchain

- Very high capacity blockchain architectures were explored by The Factoid Authority and Factom Inc. supported by the SBIR Grant DE-SC0019925 for the US Department of Energy
- Tests demonstrated that over 1 million hashes could be combined into Merkle Trees dynamically using a commodity Intel i7[1]

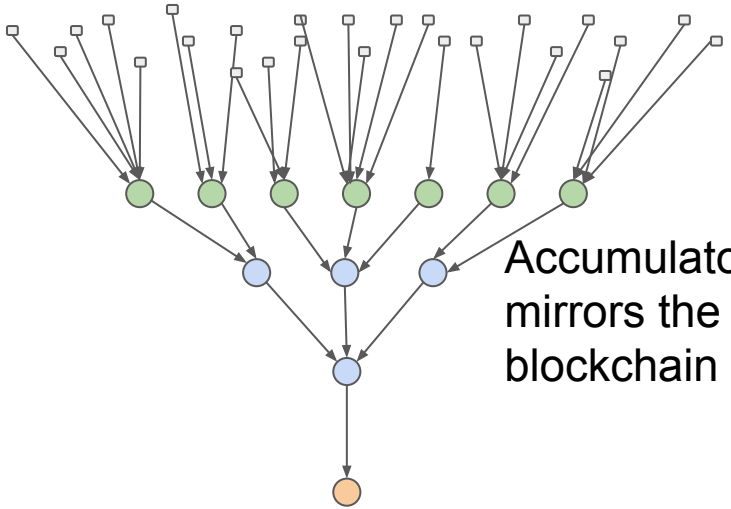
Validator / Accumulator Architecture

The construction of Merkle Trees can be done dynamically to support multiple streams of hashes from Validators



IoT Distributed Validation

Readings from devices are signed at source and routed to accumulators



Validators ensure all data is correct and properly signed.

Accumulators create the hash tree that mirrors the data flow from devices to the blockchain

The Blockchain provides cryptographic proof of the data and its organization by logging the Merkle DAG roots as they are generated

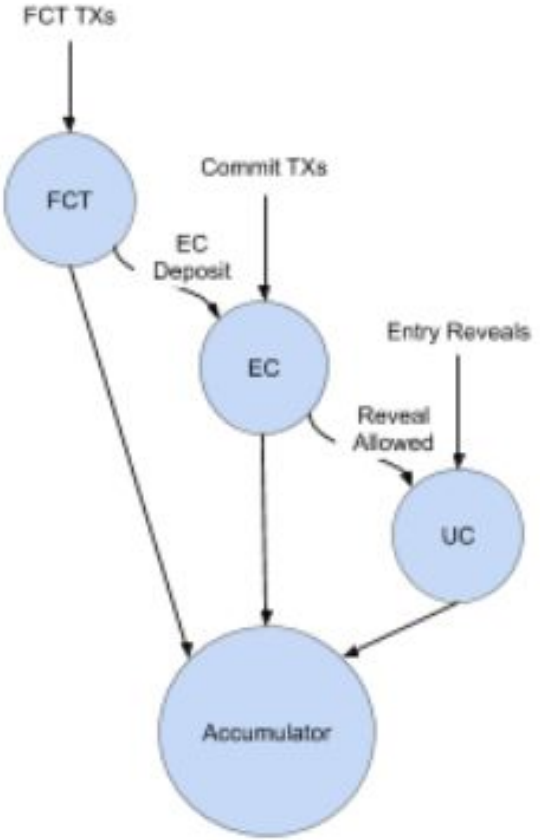
Performance Tests of Validator / Accumulator

- Low Memory/CPU requirements for Accumulation
- High performance with fast Key Value Databases
- Validation can be distributed over multiple chain types
- Validation can be executed in parallel without interaction between chain types within a block
- Chains are coordinated between blocks

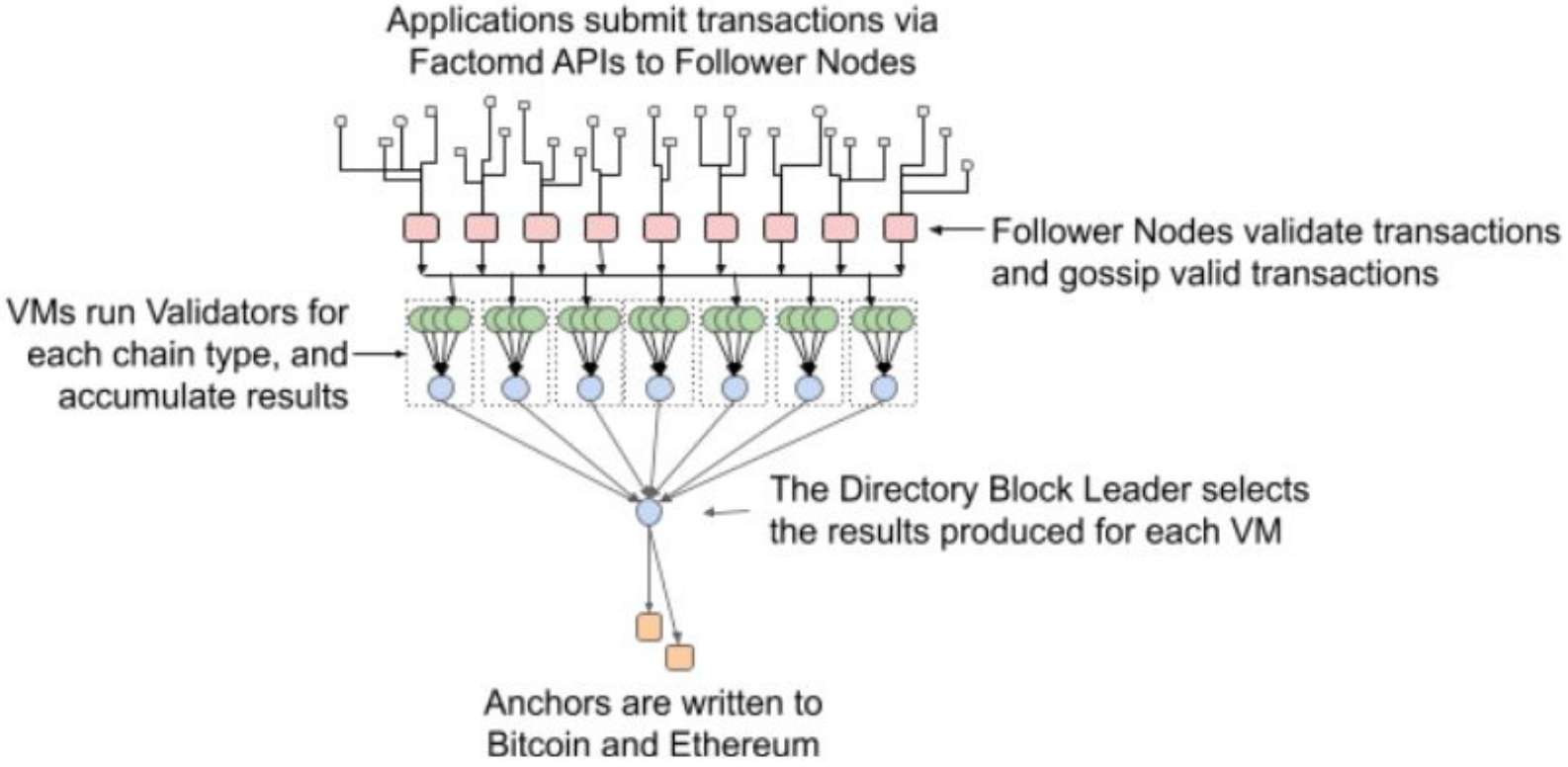
Synthetic Transactions handle chain coordination

A Blockchain can create transactions to be processed by other validators handling other chains or chain types. This allows validation to be executed at full speed, and inter chain communication to suffer minimal delays.

Synthetic Transactions shown here are the EC Deposits from converting FCT to EC, and Reveal Allowed by recording a Commit Entry.



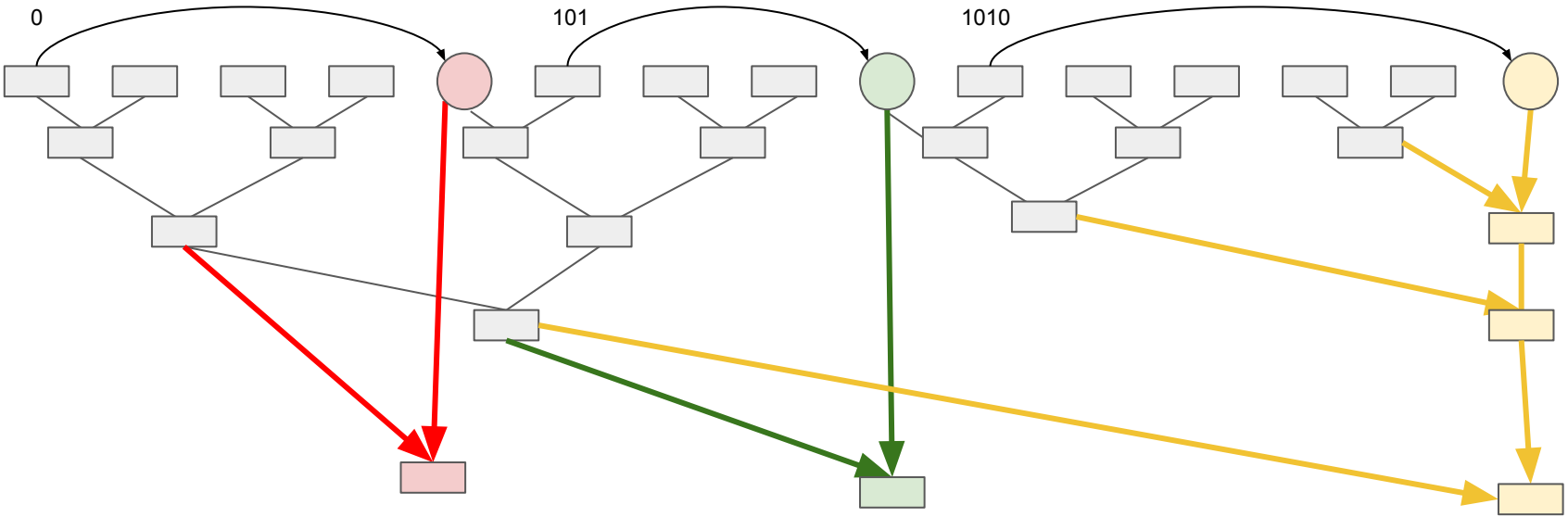
Validator / Accumulator based Factom Protocol



Stateful Merkle Trees and Merkle DAGs

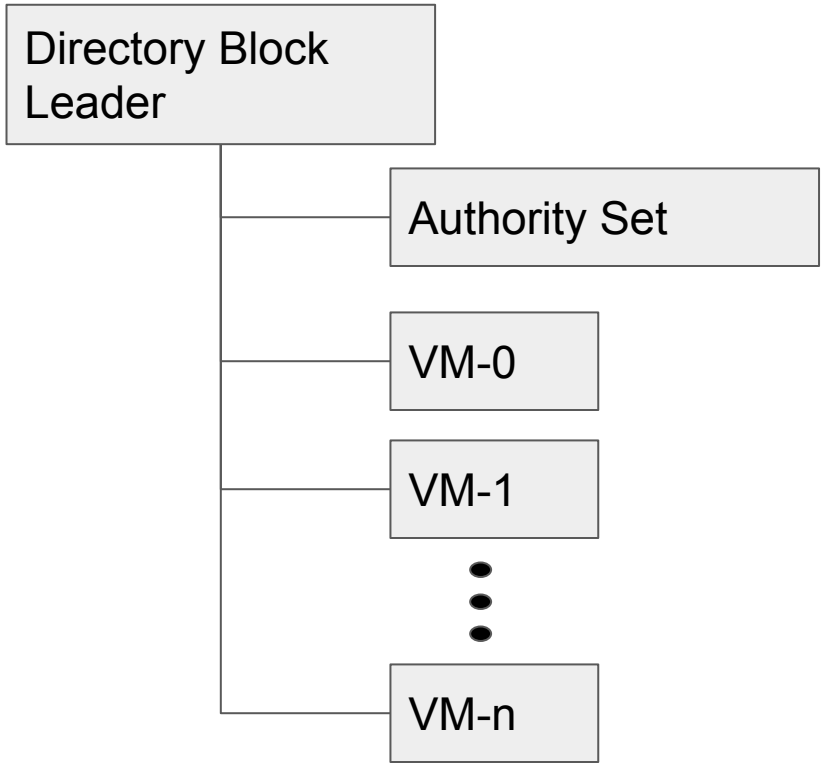
- Dynamic construction of Merkle Trees is very fast
- The state required for Dynamic construction is very small, requiring only one hash per bit of the count of entries in the Merkle Tree
- Putting headers into the Merkle Tree caches the Merkle State
- Merkle DAGs are cheaper than balancing the Merkle Tree

Merkle DAG Roots



Merkle DAG Roots can be constructed as a Merkle Tree for a user chain is constructed. Each Merkle DAG Root contains a proof of the entire user chain content that precedes it.

Move to Probabilistic Consensus



Add a Directory Block Leader. Instead of Elections, the Directory Block Leader chooses the VM solutions from the network based on priority.

Priorities are dictated by the Authority Set, also established by the Directory Block Leader

Authority Nodes validate and sign off on the Directory Block produced by the Directory Block Leader.

Layered Probabilistic Consensus

- Resolves over time (no stalls)
- Allows for competition between leaders for building the next state
- Compatible with the current Proof of Authority, but also with Proof of Stake or Proof of Work
- Adapts easily to changes in load, network conditions

Factom 2.0: Phase I

- Leaders would be refactored into the validator / accumulator architecture.
- go channels used to implement the communications between various components.
- Single server deployment of authority nodes and followers.

Estimated Performance: 1000 - 5000 tps

Factom 2.0: Phase II

- Move to multiple server deployment of factomd
- Each VM represents a shard of the network
- Followers enabled to follow selected chains

Estimated Performance: 28,000 - 144,000 tps

Factom 2.0: Phase III

- Each VM broken up into shards, with ~1000 shards total
- Shards are allocated to VMs to load balance
- Multiple shards can be run on a single system or multiple systems in response to load
- Data Servers supply data to the network

Estimated Performance: 25,000,000 - 130,000,000 tps