Catalyst Network

Developer Introduction

Created by @nshcore

What is the

Catalyst Network?

Catalyst Network

- Probabalistic BFT Consensus
- Distributed FileSystem (DFS)
- Distributed Compute System (DCS)
- Distributed DNS
- KVM smart contracts
- Public and Confidential transactions

What is

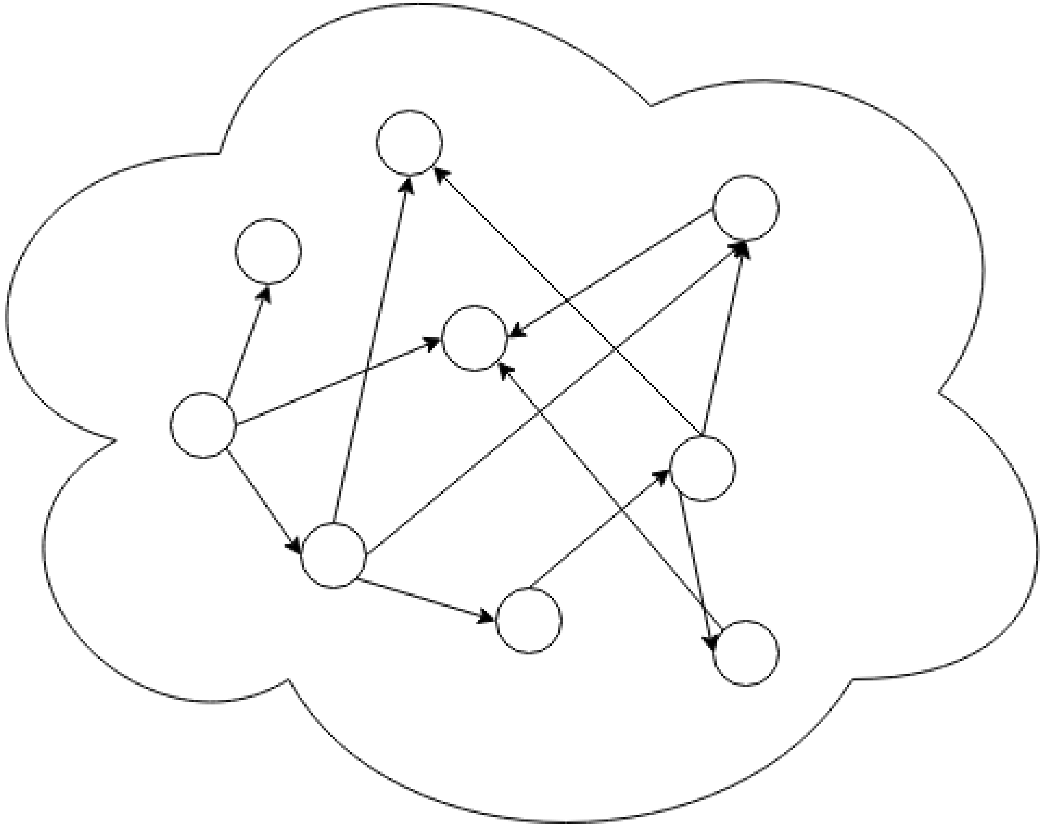
Probablistic BFT Consensus

A collabarative process in which nodes collectively vote on global state transitions

Create Transaction

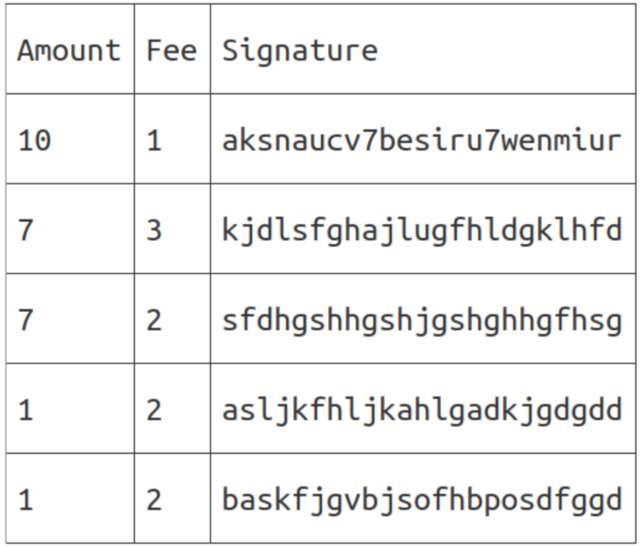
```
+---Protocol Message-------------------------------------+-----------------+
|                                                        |                 |
|   PeerID                                               | 60 bytes        |
|                                                        |                 |
+--------------------------------------------------------------------------+
|                                                        |                 |
|   CorrelationId                                        | 16 bytes        |
|                                                        |                 |
+--------------------------------------------------------------------------+
|                                                        |                 |
|   MessageType                                          | 2 bytes         |
|                                                        |                 |
+--------------------------------------------------------------------------+
|                                                        |                 |
|   Signature                                            | 64 bytes        |
|                                                        |                 |
+--------------------------------------------------------------------------+
|--Transaction Message-----------------------------------------------------|
||                     |                                 |                ||
||                     |  Public Entries                 | N * 104 bytes  ||
||                     |                                 |                ||
||                     |                                 |                ||
||                     +---------------------------------------------------|
||                     |                                 |                ||
||  Entries (N > 1)    |                                 |                ||
||                     |  Confidential Entries           | N * (104 bytes ||
||                     |                                 | + CallData)    ||
||                     |                                 |                ||
||                     +---------------------------------------------------|
||                     |                                 |                ||
||                     |                                 |                ||
||                     |  Smart Contract Entries         |  N*736 bytes   ||
||                     |                                 |                ||
||                     |                                 |                ||
|----------------------+---------------------------------------------------|
||                                                       |                ||
||  Signature                                            |  64 bytes      ||
||                                                       |                ||
|--------------------------------------------------------------------------|
||                                                       |                ||
||  Timestamp                                            |  4 bytes       ||
||                                                       |                ||
|--------------------------------------------------------------------------|
+--------------------------------------------------------------------------+

+--Public Entry----+-----------+
|                  |           |
|  Amount          |  32 bytes |
|                  |           |
+------------------------------+
|                  |           |
|  To Address      |  20 bytes |
|                  |           |
+------------------------------+
|                  |           |
|  From Address    |  20 bytes |
|                  |           |
+------------------------------+
|                  |           |
|  Transaction Fee |  32 bytes |
|                  |           |
+------------------+-----------+

+--Contract Entry--+-----------+
|                  |           |
|  Amount          |  32 bytes |
|                  |           |
+------------------------------+
|                  |           |
|  CallData        | Bytes > 0 |
|                  |           |
+------------------------------+
|                  |           |
|  To Address      |  20 bytes |
|                  |           |
+------------------------------+
|                  |           |
|  From Address    |  20 bytes |
|                  |           |
+------------------------------+
|                  |           |
|  Transaction Fee |  32 bytes |
|                  |           |
+------------------+-----------+

+--Confidential Entry-------------------+---------------+
|                                       |               |
|  value_commitment                     | M * 32 bytes  |
|                                       |               |
+-------------------------------------------------------+
|                                       |               |
|  bit_commitment                       | 32 bytes      |
|                                       |               |
+-------------------------------------------------------+
|                                       |               |
|  per_bit_blinding_factor_commitment   | 32 bytes      |
|                                       |               |
+-------------------------------------------------------+
|                                       |               |
|  poly_commitment_t1                   | 32 bytes      |
|                                       |               |
+-------------------------------------------------------+
|                                       |               |
|  poly_commitment_t2                   | 32 bytes      |
|                                       |               |
+-------------------------------------------------------+
|                                       |               |
|  proof_of_share_tau                   | 32 bytes      |
|                                       |               |
+-------------------------------------------------------+
|                                       |               |
|  proof_of_share_mu                    | 32 bytes      |
|                                       |               |
+-------------------------------------------------------+
|                                       |               |
|  aggregated_vector_polynomial_l       | k * 32 bytes  |
|                                       |               |
+-------------------------------------------------------+
|                                       |               |
|  aggregated_vector_polynomial_r       | k * 32 bytes  |
|                                       |               |
+-------------------------------------------------------+
|                                       |               |
|  a_prime_0                            | 32 bytes      |
|                                       |               |
+-------------------------------------------------------+
|                                       |               |
|  b_prime_0                            | 32 bytes      |
|                                       |               |
+---------------------------------------+---------------+
```

Broadcast to network



Deterministic Mempool

- Highest amount
- Highest fee
- Oldest timestamp
- Signature in alphabetical order



Amount	Fee	Signature
10	1	aksnaucv7besiru7wenmiur
7	3	kjdlsfghajlugfhldgklhfd
7	2	sfdhgshhgshjgshghhgfhsg
1	2	asljkfhljkahlgadkjgdgdd
1	2	baskfjgvbjsofhbposdfggd

Validation

Peers in the validation process have 4 states

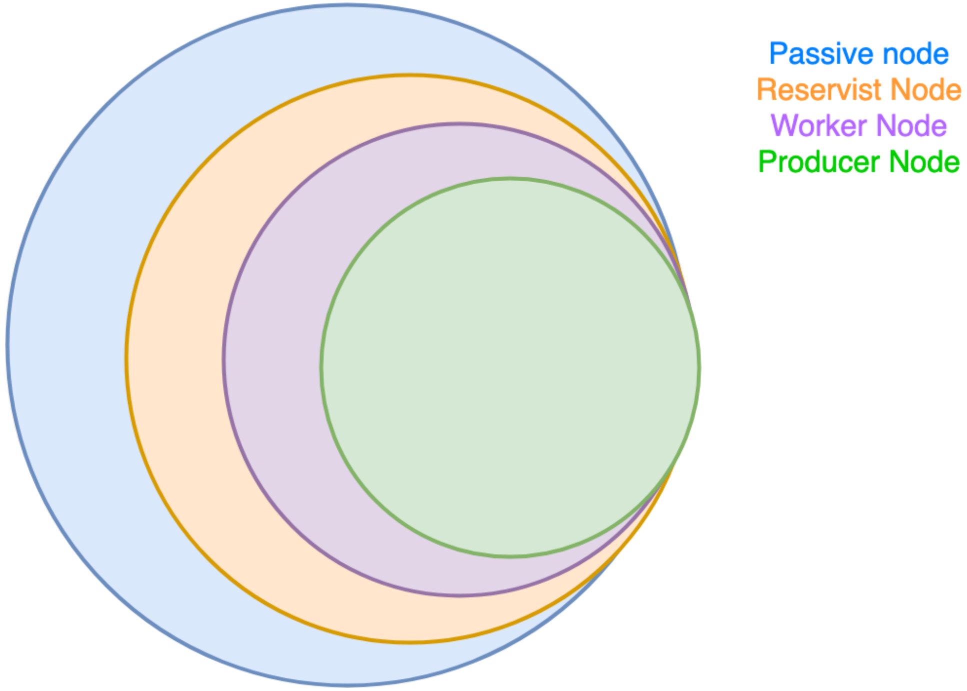
1. Passive Node
   - Connected to peer network
2. Reservist Node
   - Awaiting for admission to the validation pool
3. Worker Node
   - A node that is admited to the validation pool
4. Producer Node
   - A subset of worker nodes who can contirbute to production of ledger state updates

Producer Selection

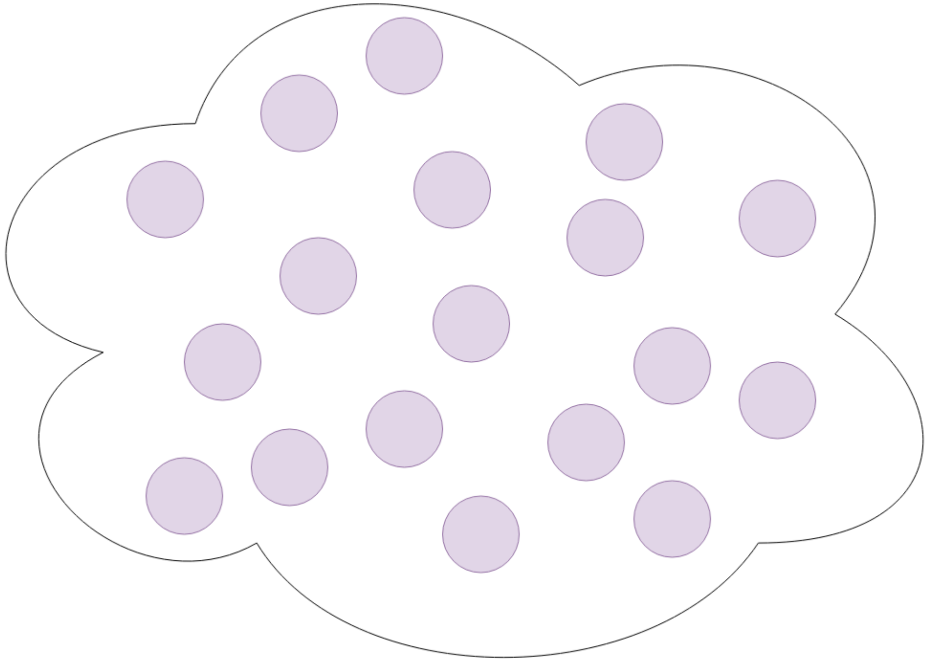
dfsHash(Δc-1) XOR PeerID

```
+-----------------------+----------------------+-----------------------+------------------------+----------------------+-----------------------+
|                       |                      |                       |                        |                      |                       |
|                       |                      |                       |                        |                      |                       |
|  dfsHash(Δc−1)        |  dfsHash(Δc−1)       |  dfsHash(Δc−1)        |  dfsHash(Δc−1)         |  dfsHash(Δc−1)       |  dfsHash(Δc−1)        |
|                       |                      |                       |                        |                      |                       |
|  cycle 2              |  cycle 3             |  cycle 4              |  cycle 5               |  cycle 6             |  cycle 7              |
|                       |                      |                       |                        |                      |                       |
+-----------------------+----------------------+-----------------------+------------------------+----------------------+-----------------------+
```

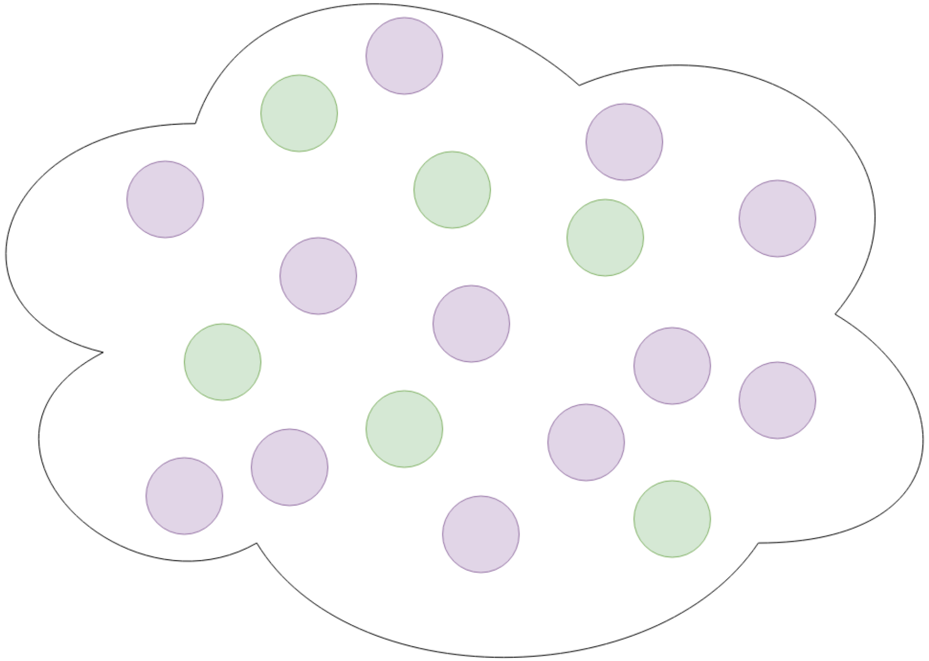
Network Constitution



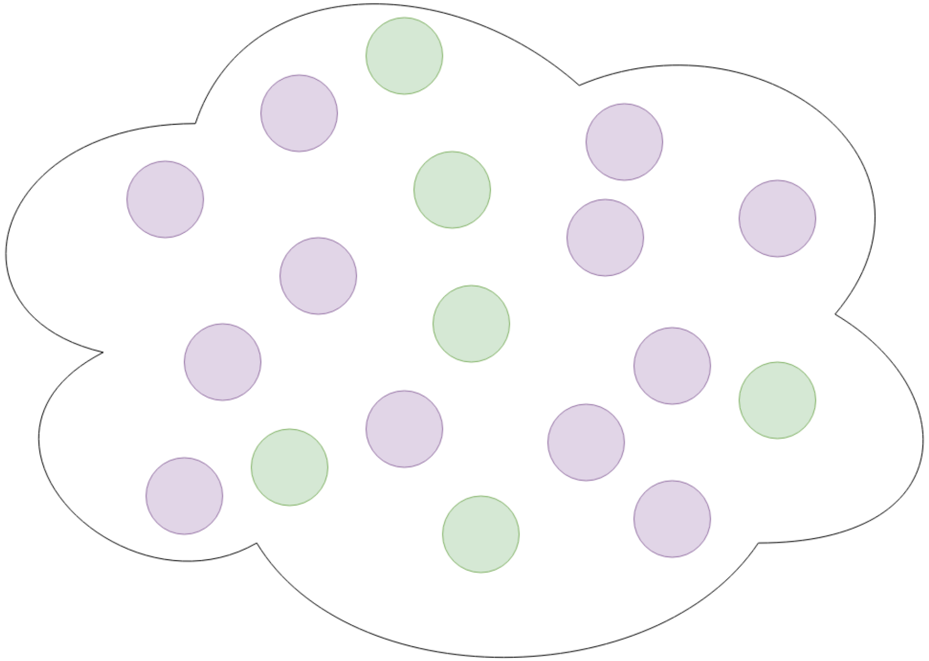
Worker pool



Worker pool

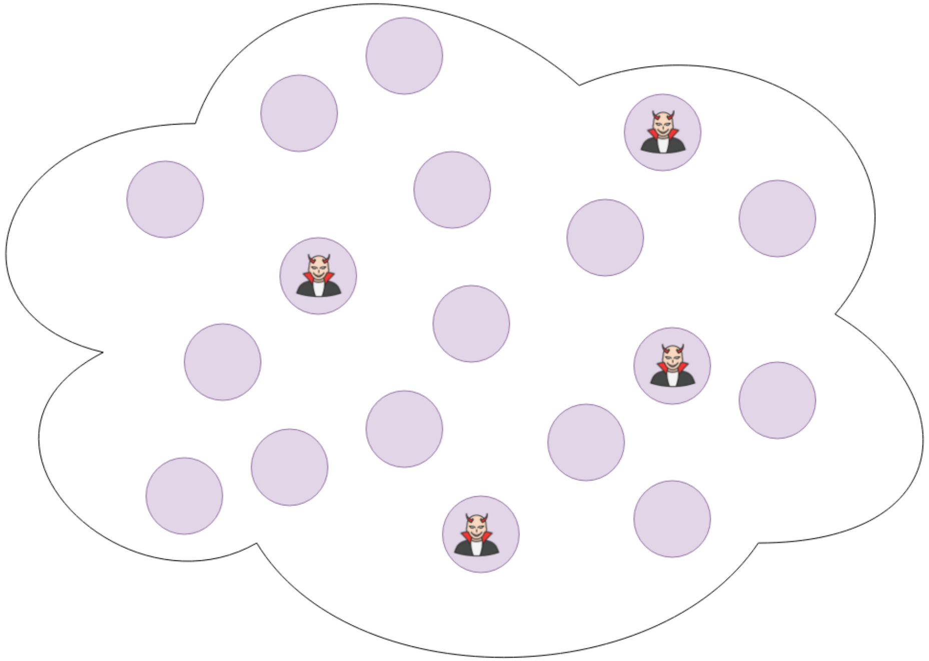


Worker pool

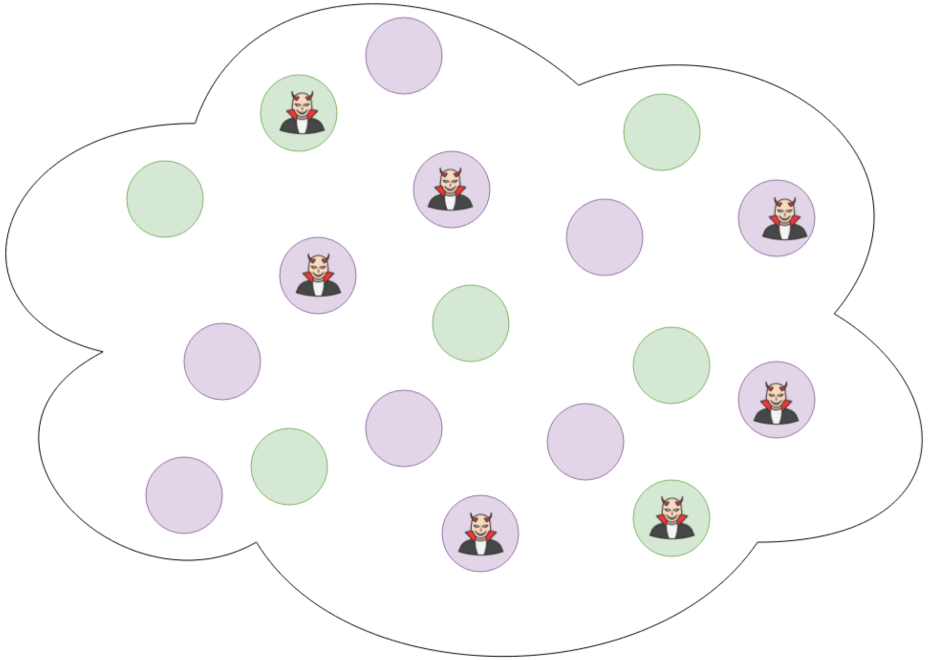


Evil Nodes

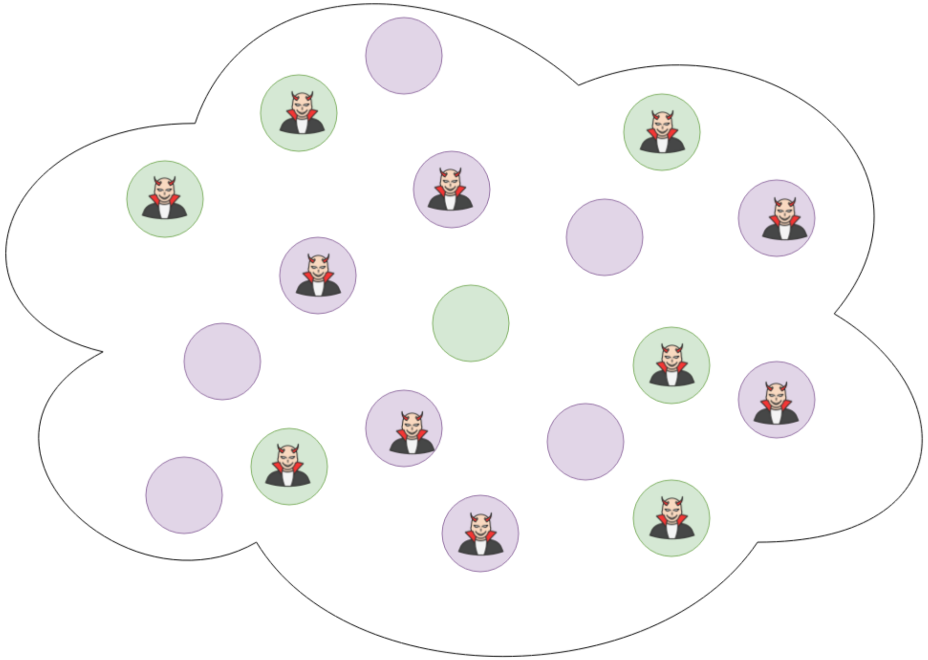
Worker pool



Worker pool



Worker pool

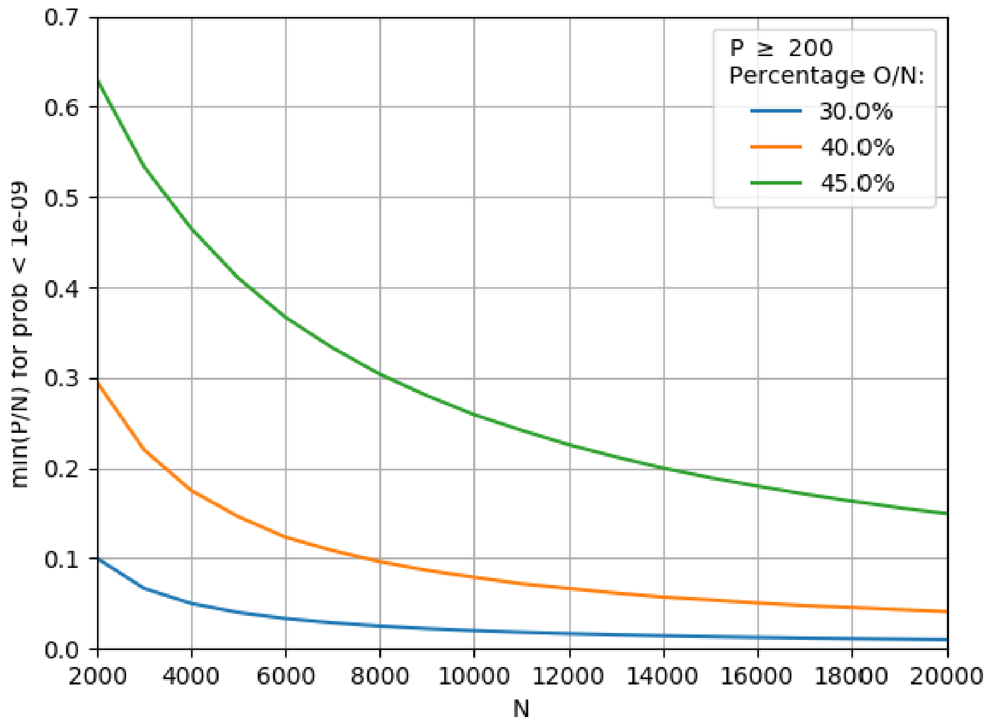


SAD BANANA
IS SAD
Text Memes .com

51% Attack Research

$$P_A = \sum_{p=p_0}^{V} P_a(p)$$

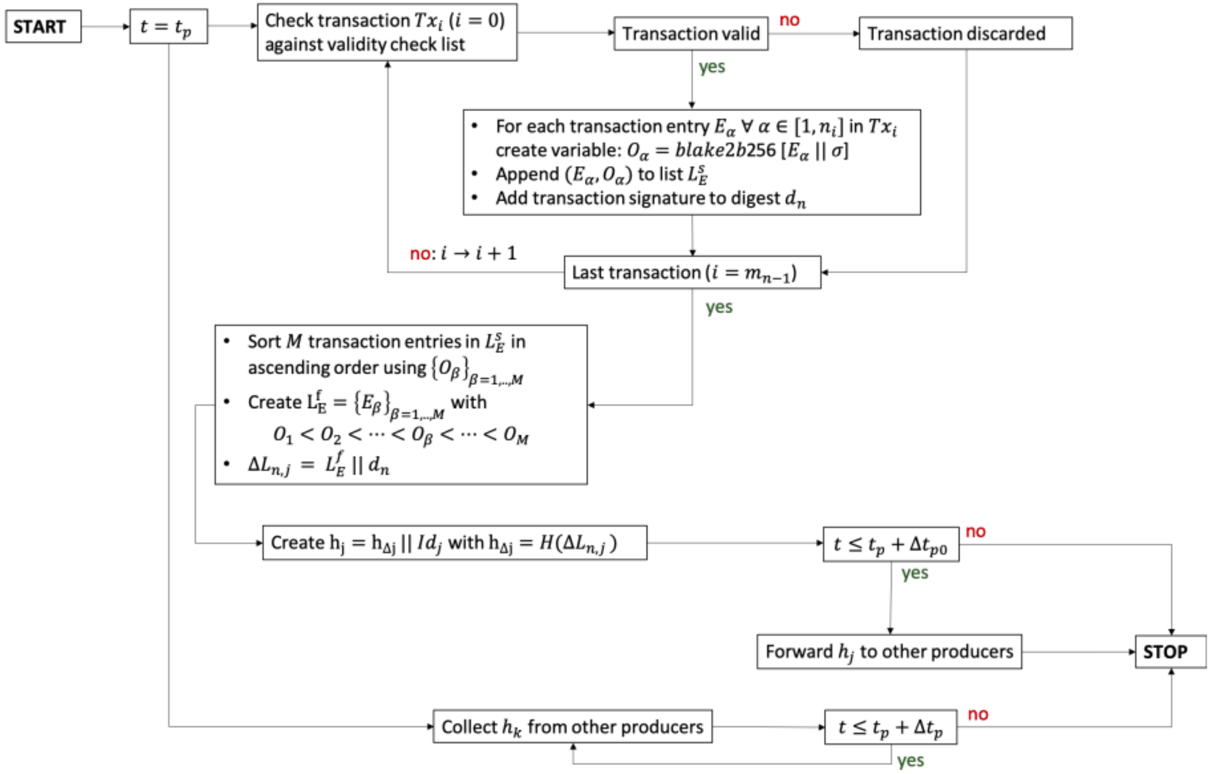
http://github.com/catalyst-network/51-percent-attack-research



The Ledger Cycle

- Producers validate common sets of transactions from the mempool
- Each producer compiles a state delta and votes among its peers in the cycle to vote on the most popular delta produced by the set of producers.

The Ledger Cycle



START

$t = t_p$

Check transaction Tx_i $(i = 0)$ against validity check list

Transaction valid

no

Transaction discarded

yes

- For each transaction entry E_α $\forall$ $\alpha \in [1, n_i]$ in Tx_i create variable: $O_\alpha = blake2b256\,[E_\alpha\,||\,\sigma]$
- Append (E_α, O_α) to list L_E^s
- Add transaction signature to digest d_n

no: $i \rightarrow i + 1$

Last transaction $(i = m_{n-1})$

yes

- Sort M transaction entries in L_E^s in ascending order using $\{O_\beta\}_{\beta=1,..,M}$
- Create $\mathrm{L_E^f} = \{E_\beta\}_{\beta=1,..,M}$ with $O_1 < O_2 < \cdots < O_\beta < \cdots < O_M$
- $\Delta L_{n,j} = L_E^f\,||\,d_n$

Create $\mathrm{h_j} = \mathrm{h_{\Delta j}}\,||\,Id_j$ with $\mathrm{h_{\Delta j}} = H(\Delta L_{n,j})$

$t \leq t_p + \Delta t_{p0}$

no

yes

Forward h_j to other producers

STOP

Collect h_k from other producers

$t \leq t_p + \Delta t_p$

no

yes

State Update

When producers have voted and come to consensus on the most correct state update, a state delta update is broadcast to the rest of the network

```
 message Delta {
        bytes previous_delta_dfs_hash = 1;
        bytes state_trie = 2;
        bytes receipt_trie = 3;
        google.protobuf.Timestamp time_stamp = 4;
        repeated Transaction.PublicEntry public_entries = 5;
        repeated Transaction.ConfidentialEntry confidential_entries = 6;
        repeated Transaction.ContractEntry contract_entries = 7;
        repeated Transaction.CoinbaseEntry coinbase_entries = 8;
}
```

Who then clean the validated tx's from their mempool

Consensus Cycle

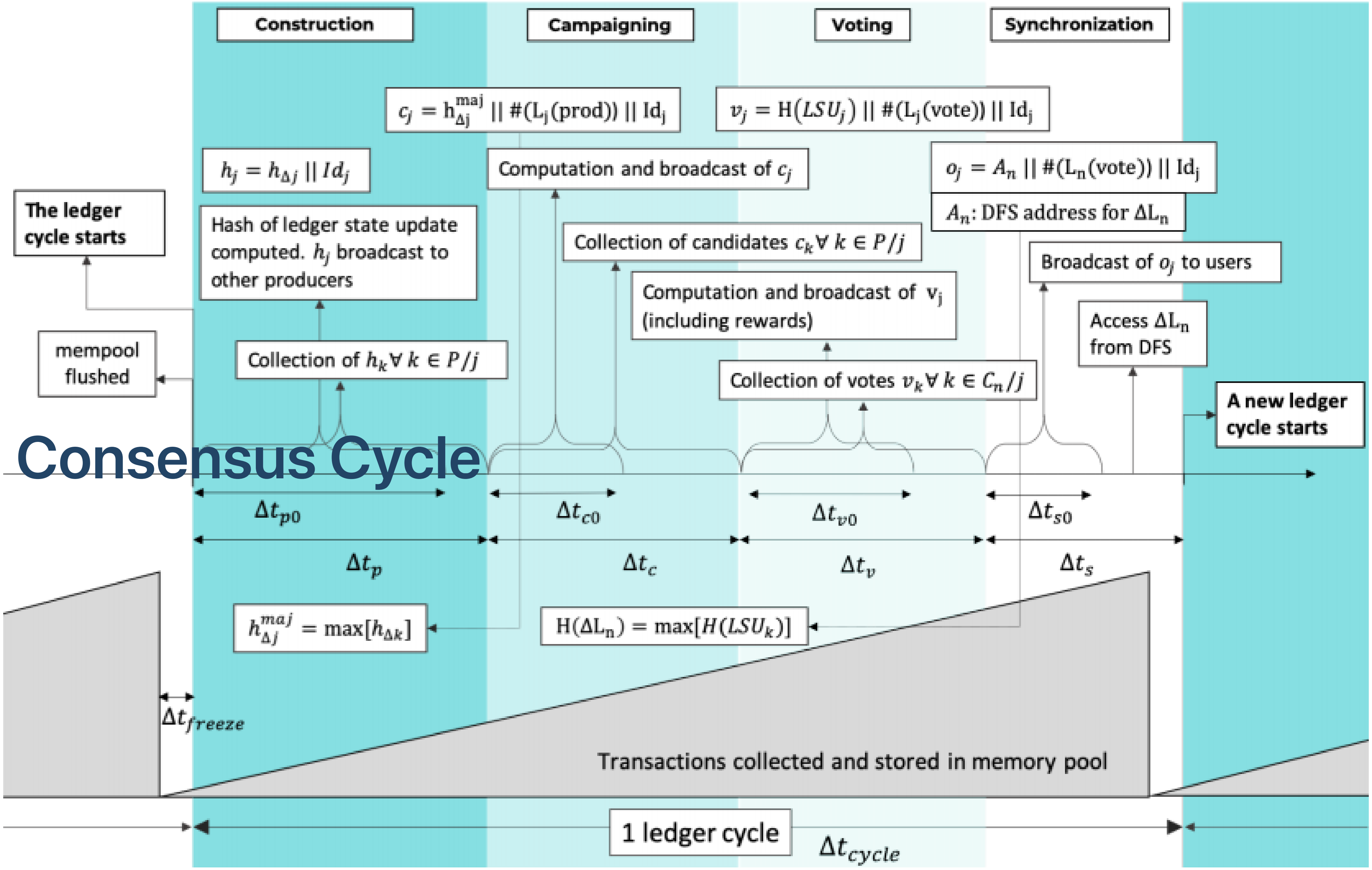

Construction
Campaigning
Voting
Synchronization
$c_j = h_{\Delta j}^{maj} || \#(L_j(prod)) || Id_j$
$v_j = H(LSU_j) || \#(L_j(vote)) || Id_j$
$h_j = h_{\Delta j} || Id_j$
Computation and broadcast of c_j
$o_j = A_n || \#(L_n(vote)) || Id_j$
A_n: DFS address for ΔL_n
The ledger cycle starts
Hash of ledger state update computed. h_j broadcast to other producers
Collection of candidates $c_k \forall k \in P/j$
Broadcast of o_j to users
Computation and broadcast of v_j (including rewards)
Access ΔL_n from DFS
mempool flushed
Collection of $h_k \forall k \in P/j$
Collection of votes $v_k \forall k \in C_n/j$
A new ledger cycle starts
Δt_{p0}
Δt_{c0}
Δt_{v0}
Δt_{s0}
Δt_p
Δt_c
Δt_v
Δt_s
$h_{\Delta j}^{maj} = \max[h_{\Delta k}]$
$H(\Delta L_n) = \max[H(LSU_k)]$
Δt_{freeze}
Transactions collected and stored in memory pool
1 ledger cycle
Δt_{cycle}

Figure 5.5: Illustration of the different phases followed by a producer during a ledger cycle.

Catalyst.Core Framework

Architecture

```
+---------------------------+--------------------------+
|                           |                          |
|                           |                          |
|     Catalyst.Modules      |     Custom.Modules.*     |
|                           |                          |
+---------------------------+--------------------------+
|                                                      |
|                                                      |
|     Catalyst.Core.Modules.*                          |
|                                                      |
+------------------------------------------------------+
|                                                      |
|                                                      |
|     Catalyst.Core.Lib                                |
|                                                      |
+------------------------------------------------------+-------------------------+----------------------------+
|                                                      |                         |                            |
|                                                      |                         |                            |
|     Protocol.Sdk.C#                                  |     Protocol.Sdk.Js     |     Protocol.Sdk.Java      |
|                                                      |                         |                            |
+------------------------------------------------------+-------------------------+----------------------------+
|                                                                                                             |
|                                                                                                             |
|     Catalyst.Protocol                                                                                       |
|                                                                                                             |
+-------------------------------------------------------------------------------------------------------------+
```

Catalyst.Protocol

- Abstract Protocol Schema Deffinitions.
- Protobuffs Serialisation Format.
- Language Interoperability.

Protocol-sdk-*

- Concrete standardised types across languages
- Auto generated from protobuffs deffinitions
- Building blocks for low level functionality

Catalyst.Core.Lib

Core librarys and helpers for IO, networking and configuration. Low level functionality for nodes and modules

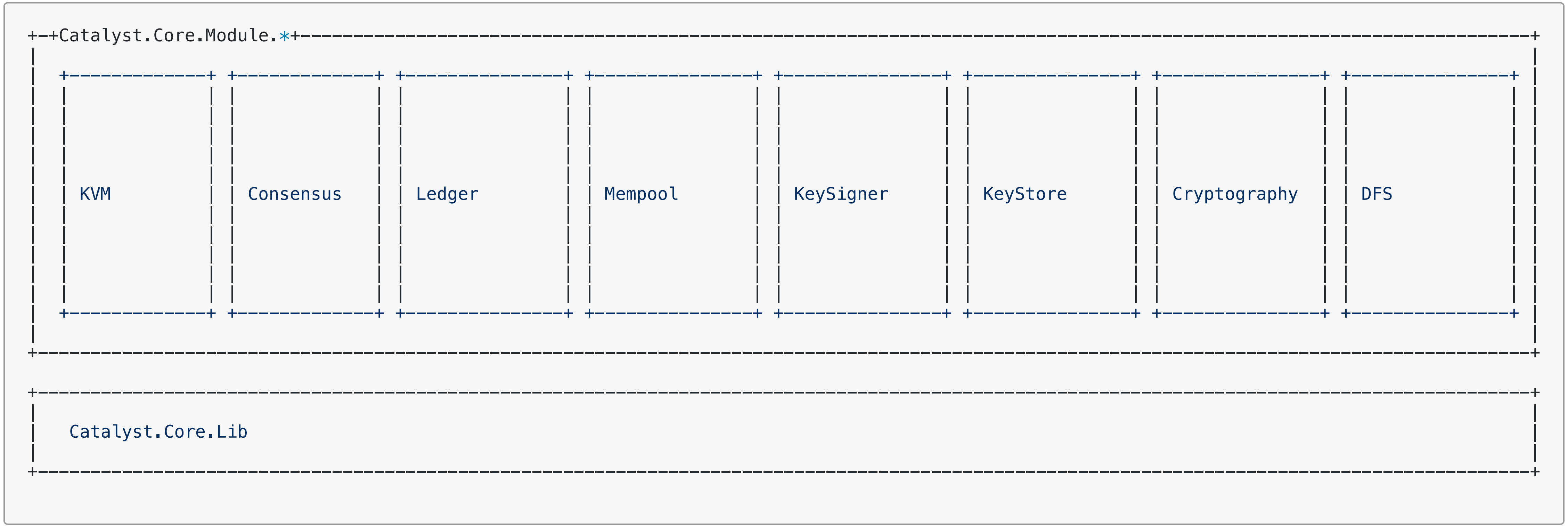
- IO
- P2P
- Cryptography

Catalyst.Core.Modules.*

- Core modules provide spoecific functionality for Catalyst.Node

Catalyst.Modules.*

- Modules provide 'flavour' functionality for nodes
- Use familiar technology
- Extend core functionality



Catalyst.Core.Module.*

KVM

Consensus

Ledger

Mempool

KeySigner

KeyStore

Cryptography

DFS

Catalyst.Core.Lib

Custom modules

Custom modules are a great way of providing additional or domain specific functionality to the node.

Catalyst.Node.POA-CE

```
+--+Catalyst.Node.POA-CE+-------------------------------------------------------------------------------------------+
|                                                                                                                   |
|  +----------------+ +---------------+                                                                             |
|  |                | |               |                                                                             |
|  |                | |               |                                                                             |
|  | POA.Consensus  | | POA.Discovery |                                                                             |
|  |                | |               |                                                                             |
|  |                | |               |                                                                             |
|  |                | |               |                                                                             |
|  +----------------+ +---------------+                                                                             |
|                                                                                                                   |
| +--+Catalyst.Core.Modules+--------------------------------------------------------------------------------------+ |
| |                                                                                                               | |
| | +---------------+ +---------------+ +---------------+ +---------------+ +---------------+  +---------------+  | |
| | |               | |               | |               | |               | |               |  |               |  | |
| | |               | |               | |               | |               | |               |  |               |  | |
| | |               | |               | |               | |               | |               |  |               |  | |
| | |               | |               | |               | |               | |               |  |               |  | |
| | |               | |               | |               | |               | |               |  |               |  | |
| | | Ledger        | | Mempool       | | KeySigner     | | KeyStore      | | Cryptography  |  |  Dfs          |  | |
| | |               | |               | |               | |               | |               |  |               |  | |
| | |               | |               | |               | |               | |               |  |               |  | |
| | |               | |               | |               | |               | |               |  |               |  | |
| | |               | |               | |               | |               | |               |  |               |  | |
| | |               | |               | |               | |               | |               |  |               |  | |
| | +---------------+ +---------------+ +---------------+ +---------------+ +---------------+  +---------------+  | |
| |                                                                                                               | |
| +---------------------------------------------------------------------------------------------------------------+ |
|                                                                                                                   |
| +---------------------------------------------------------------------------------------------------------------+ |
| |                                                                                                               | |
| |  Catalyst.Core.Lib                                                                                            | |
| |                                                                                                               | |
| +---------------------------------------------------------------------------------------------------------------+ |
+-------------------------------------------------------------------------------------------------------------------+
```

Catalyst.Node-CE

```
+--Catalyst.Node-CE--------------------------------------------------------------------------------------------------------------------------------+
| +---------------+ +---------------+  +---------------+                                                                                           |
| |               | |               |  |               |                                                                                           |
| |               | |               |  |               |                                                                                           |
| |               | |               |  |               |                                                                                           |
| |               | |               |  |               |                                                                                           |
| |               | |               |  |               |                                                                                           |
| | DCS           | | Web3          |  | DDNS          |                                                                                           |
| |               | |               |  |               |                                                                                           |
| |               | |               |  |               |                                                                                           |
| |               | |               |  |               |                                                                                           |
| |               | |               |  |               |                                                                                           |
| |               | |               |  |               |                                                                                           |
| +---------------+ +---------------+  +---------------+                                                                                           |
|                                                                                                                                                  |
| +-+Catalyst.Core.Module.*+---------------------------------------------------------------------------------------------------------------------+ |
| |                                                                                                                                              | |
| |  +-------------+ +-------------+ +---------------+ +---------------+ +---------------+ +---------------+ +---------------+ +---------------+ | |
| |  |             | |             | |               | |               | |               | |               | |               | |               | | |
| |  |             | |             | |               | |               | |               | |               | |               | |               | | |
| |  |             | |             | |               | |               | |               | |               | |               | |               | | |
| |  |             | |             | |               | |               | |               | |               | |               | |               | | |
| |  |             | |             | |               | |               | |               | |               | |               | |               | | |
| |  | KVM         | | Consensus   | | Ledger        | | Mempool       | | KeySigner     | | KeyStore      | | Cryptography  | | DFS           | | |
| |  |             | |             | |               | |               | |               | |               | |               | |               | | |
| |  |             | |             | |               | |               | |               | |               | |               | |               | | |
| |  |             | |             | |               | |               | |               | |               | |               | |               | | |
| |  |             | |             | |               | |               | |               | |               | |               | |               | | |
| |  |             | |             | |               | |               | |               | |               | |               | |               | | |
| |  +-------------+ +-------------+ +---------------+ +---------------+ +---------------+ +---------------+ +---------------+ +---------------+ | |
| |                                                                                                                                              | |
| +----------------------------------------------------------------------------------------------------------------------------------------------+ |
|                                                                                                                                                  |
| +----------------------------------------------------------------------------------------------------------------------------------------------+ |
| |                                                                                                                                              | |
| |   Catalyst.Core.Lib                                                                                                                          | |
| |                                                                                                                                              | |
| +----------------------------------------------------------------------------------------------------------------------------------------------+ |
+--------------------------------------------------------------------------------------------------------------------------------------------------+
```

KVM

ERC Standards on KVM

KVM is designed to be a host for ERC-20, ERC-223, ERC-721 and other standard Ethereum contracts

Ethereum DeFi on KVM

With EVM compatibility KVM can bring successful Ethereum solutions like UniSwap or MakerDAO into Catalyst and allow users to interact with DeFi tools they feel familiar with

Integration with Ethereum tools

KVM developers will be able to use Solidity, Vyper and Truffle for contract development and deployments.KVM will create a mapping between known Web3 / JSON RPC calls and Catalyst state.

Native DFS access from KVM

Catalyst users will have a unique ability to build KVM contracts that will have read and write access to big data sets stored on DFS.Prefetch mechanisms and local DFS caches deliver a solution to data availability for virtual machines.

KVM with Catalyst cryptography

KVM will extend EVM allowing Catalyst users to create and verify Catalyst cryptographical artifacts.Catalyst signatures and multihashes will be available for KVM smart contract developers.

State rent mechanism

KVM will be from the beginning balancing the cost of local, high availability state and the long term DFS storage.State rent and DFS cache mechanism will allow to correctly attribute storage cost to contract owners and users.

Early Access Program

Sign up to the Catalyst Network EAP

https://catalystnet.org/eap

Thanks

twitter.com/nshcore
github.com/nshcore