DAGGER
BY THE COMMUNITY

XDAG: PoW + DAG

frozen@xdag.io

DAGGER
BY THE COMMUNITY

XDAG: A new DAG-based cryptocurrency

The first mineable DAG

No Pre-mine

No ICO

Community driven

DAGGER
BY THE COMMUNITY

State-of-The-Art

Bitcoin

Ethereum

EOS

???

DAGGER
BY THE COMMUNITY

Features:

- PoW
- High TPS
- Decentralized
- Block = Transaction = Address

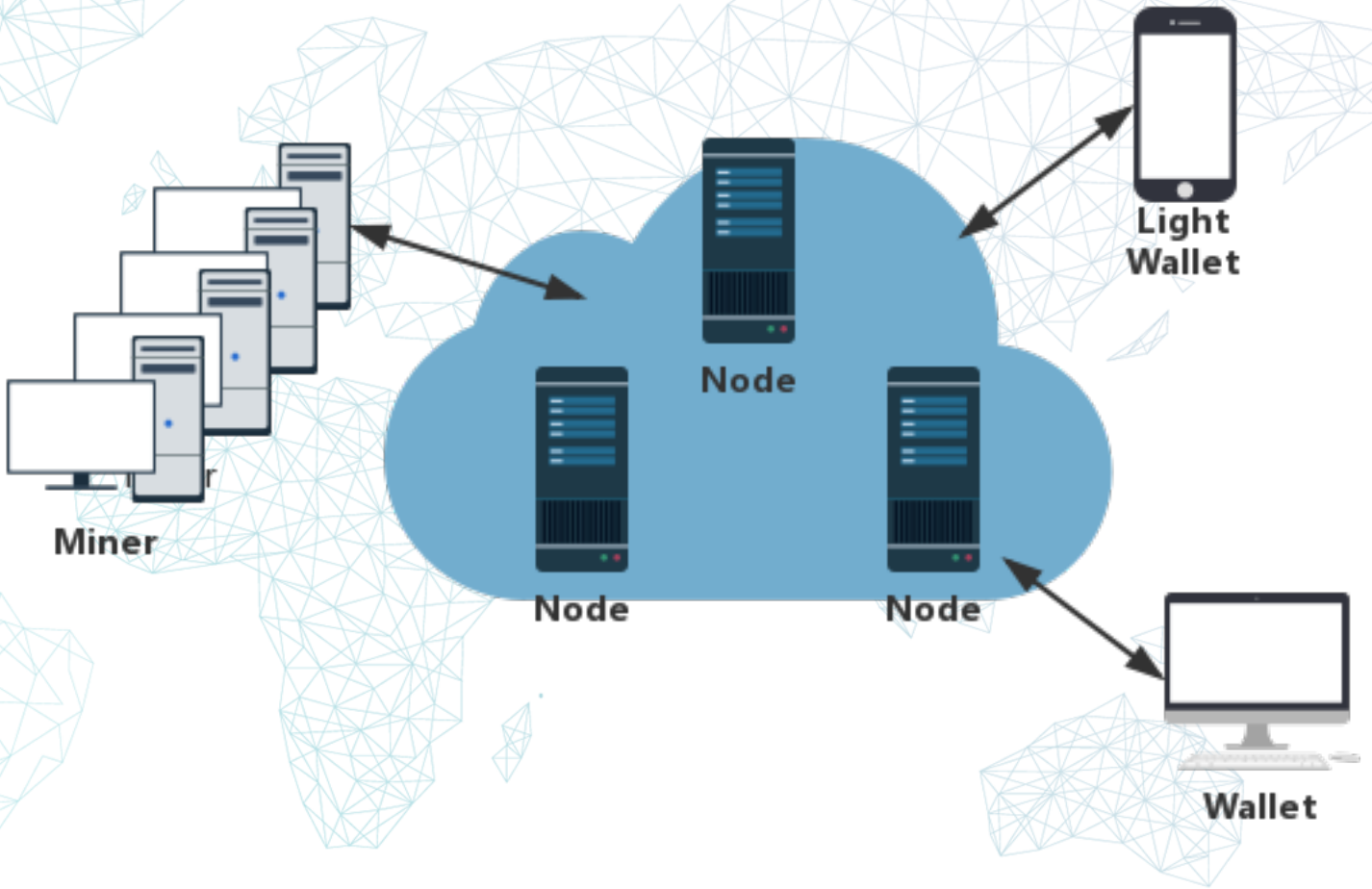
DAGGER
BY THE COMMUNITY

- Why PoW
- Why DAG
- How high TPS

DAGGER
BY THE COMMUNITY

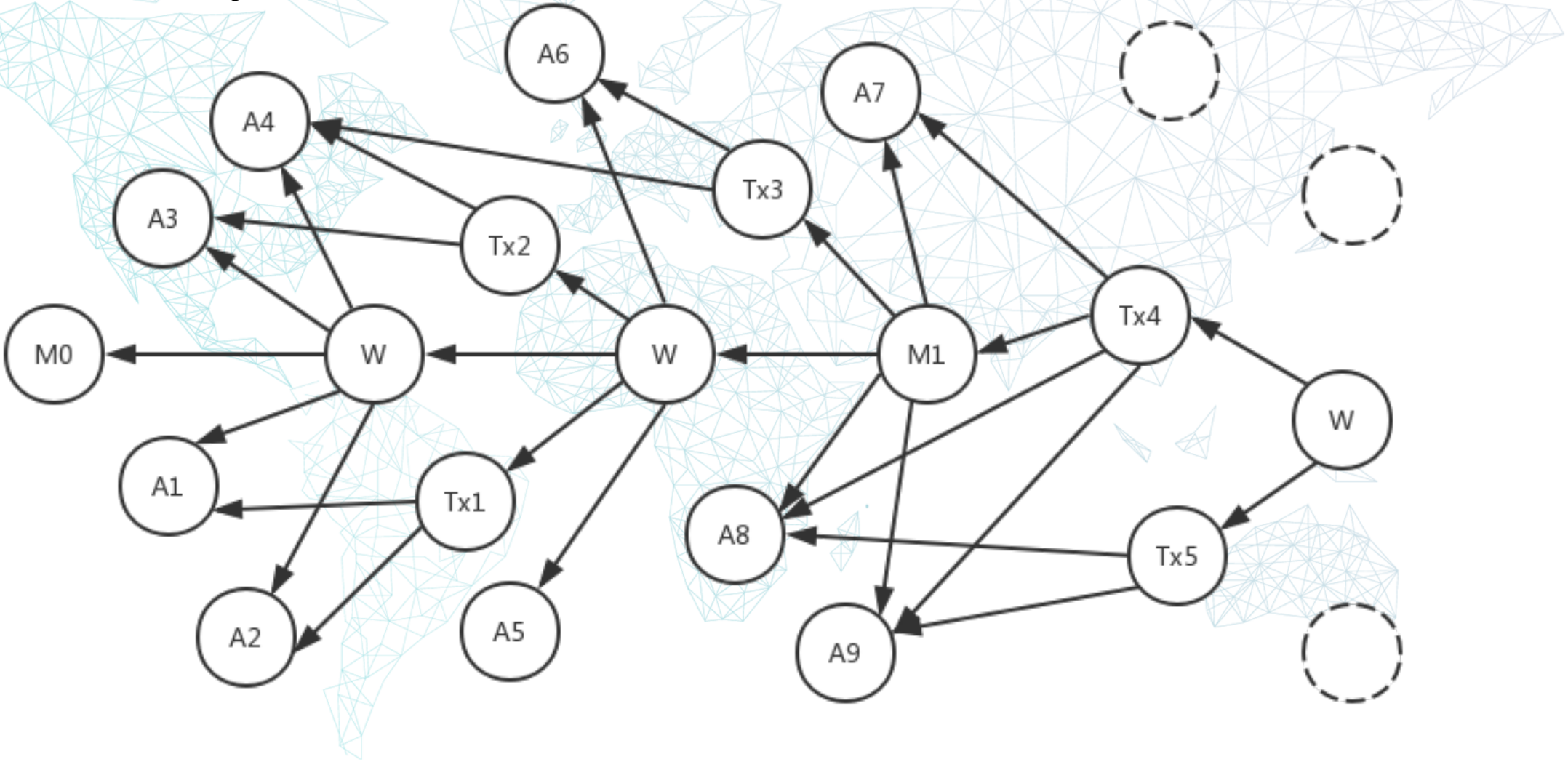
Topology

- Node (Pool)
- Wallet / CPU Miner
- GPU Miner



DAGGER
BY THE COMMUNITY

XDAG Simple Case



DAGGER
BY THE COMMUNITY

Block

persistent storage

- 512 Bytes
- 5 Forms
- 16 types
- 16 fields

```
#define XDAG_BLOCK_FIELDS 16

typedef uint64_t xdag_time_t;
typedef uint64_t xdag_amount_t;
typedef uint64_t xdag_hash_t[4];
typedef uint64_t xdag_hashlow_t[3];

struct xdag_field {
    union {
        struct {
            union {
                struct {
                    uint64_t transport_header;
                    uint64_t type;
                    xdag_time_t time;
                };
                xdag_hashlow_t hash;
            };
            union {
                xdag_amount_t amount;
                xdag_time_t end_time;
            };
        };
        xdag_hash_t data;
    };
};

struct xdag_block {
    struct xdag_field field[XDAG_BLOCK_FIELDS];
};
```

DAGGER
BY THE COMMUNITY

Block example

- 512 Bytes
- 1 header
- 15 fields
- Storage on disk
- max limit 12 Txs

8 Bytes	8 Bytes	8 Bytes	8 Bytes
transport header	type	time	amount
Output1 hash			amount
Input1 hash			amount
Input2 hash			amount
Input3 hash			amount
Input4 hash			amount
Input5 hash			amount
Input6 hash			amount
Input7 hash			amount
Public Key 1			
Input sign R 1			
Input sign S 1			
Public Key 2			
Input sign R 2			
Input sign S 2			

DAGGER
BY THE COMMUNITY

Internal Block

- store DAG
- store block info

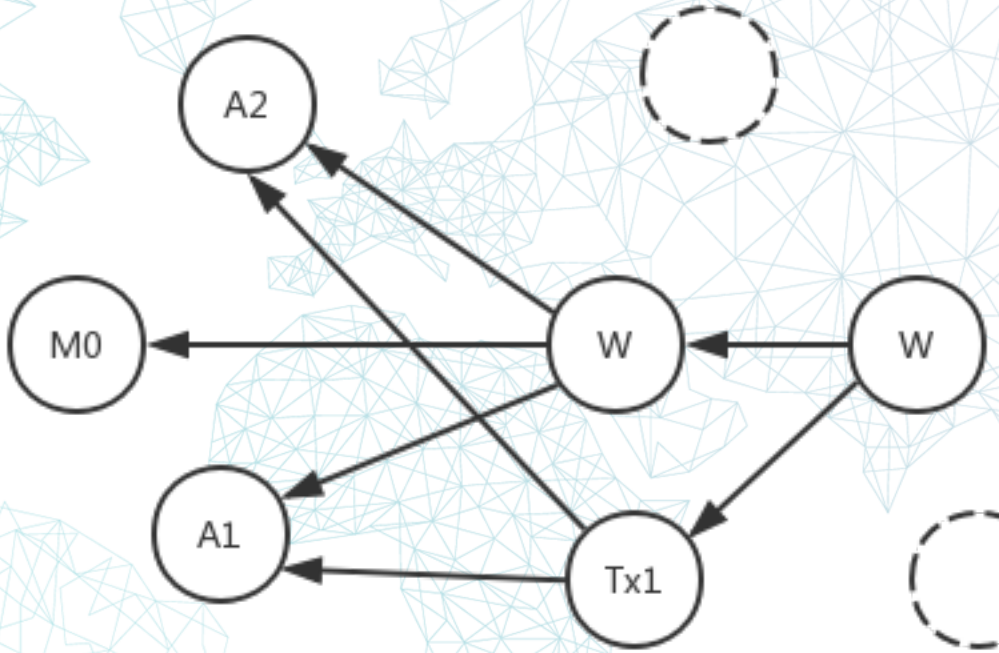
```
struct block_backrefs {
    struct block_internal *backrefs[N_BACKREFS];
    struct block_backrefs *next;
};

struct block_internal {
    struct ldus_rbtree node;
    xdag_hash_t hash;
    xdag_diff_t difficulty;
    xdag_amount_t amount, linkamount[MAX_LINKS], fee;
    xdag_time_t time;
    uint64_t storage_pos;
    struct block_internal *ref, *link[MAX_LINKS];
    struct block_backrefs *backrefs;
    uint8_t flags, nlinks, max_diff_link, reserved;
    uint16_t in_mask;
    uint16_t n_our_key;
};
```

DAGGER
BY THE COMMUNITY

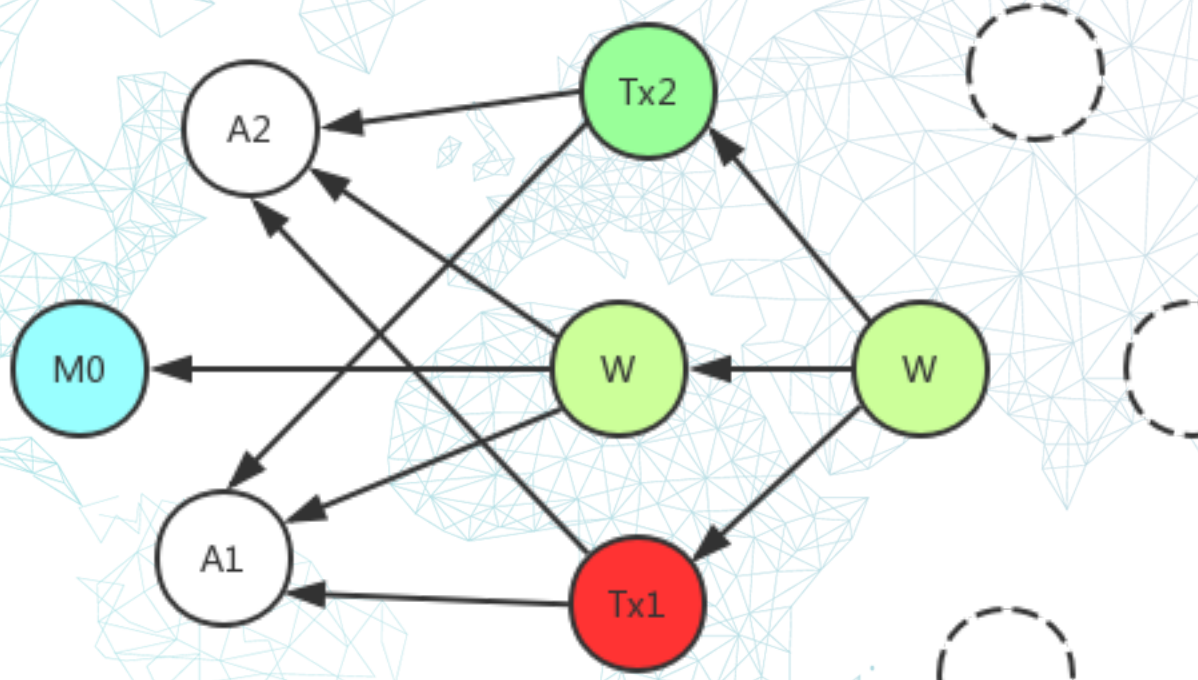
Simple Transaction case

- A1 A2 address
- M0 main block
- Tx1 transaction
- W witness block



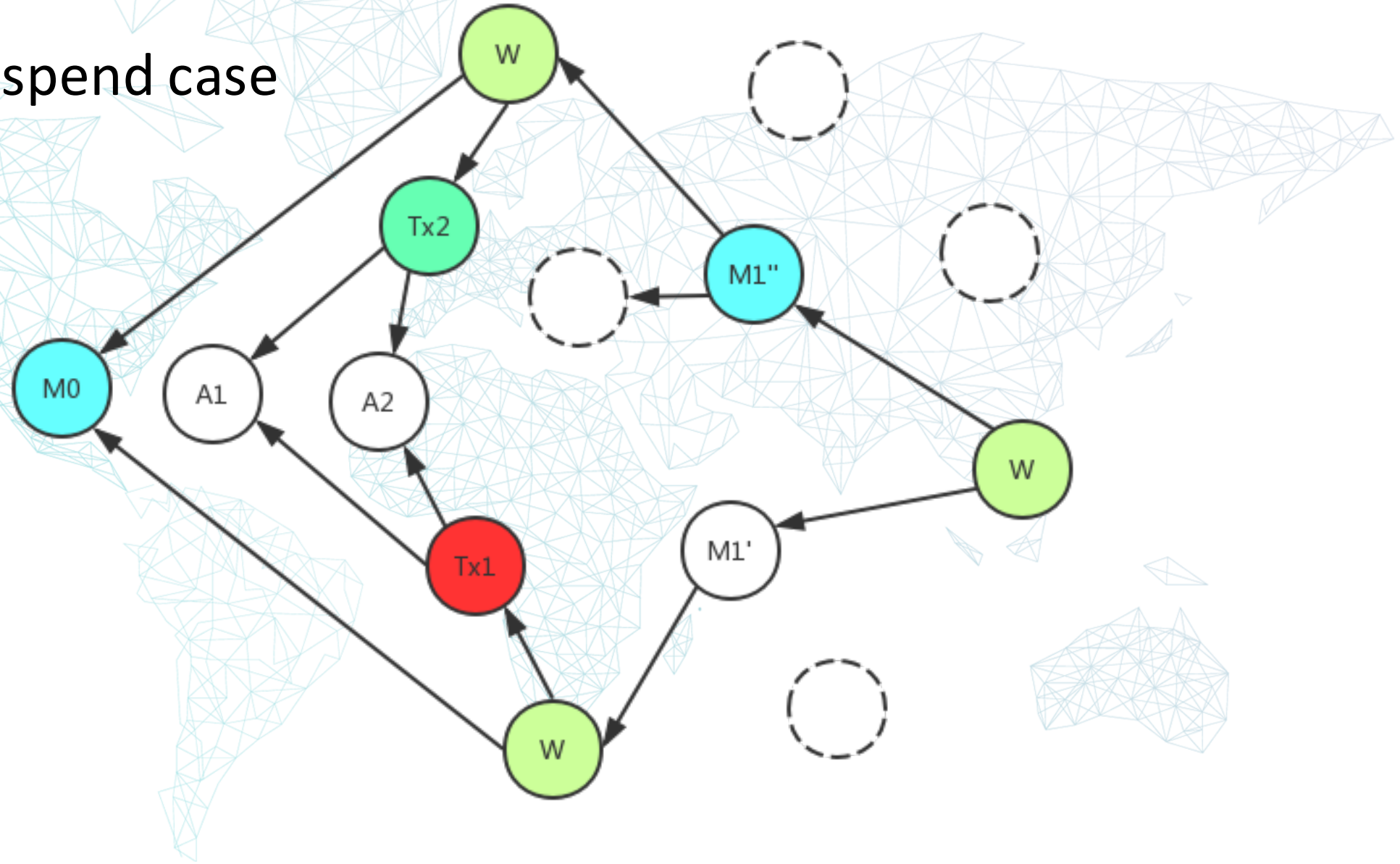
DAGGER
BY THE COMMUNITY

Double spend case



DAGGER
BY THE COMMUNITY

Double spend case



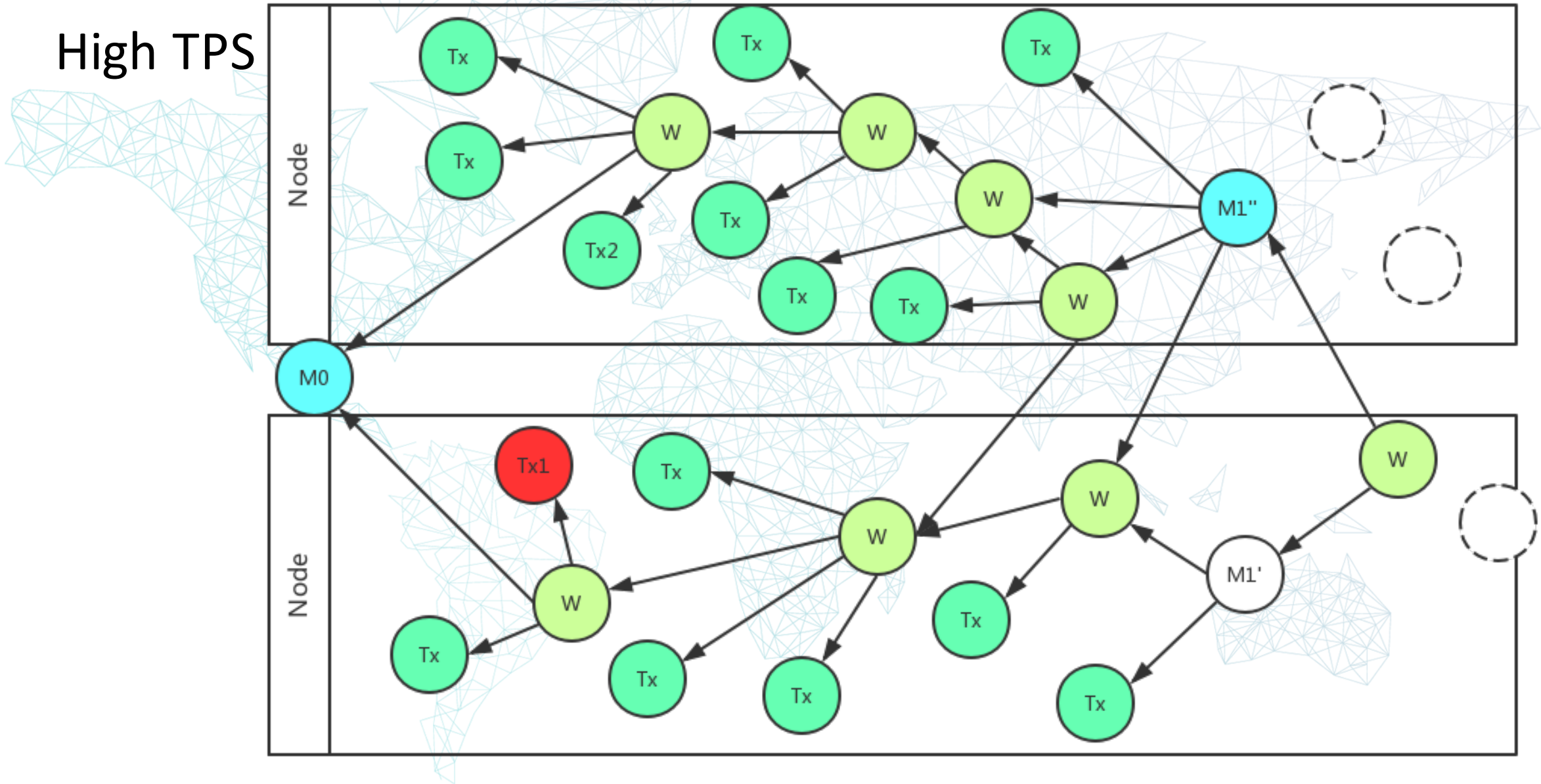
DAGGER
BY THE COMMUNITY

PoW

- Miner use sha256d to find minimal hash
- Node generates main block based on minimal hash every 64s
- Main net use generated main block to determine main chain

DAGGER
BY THE COMMUNITY

High TPS



DAGGER
BY THE COMMUNITY

Algorithm : how to validate transaction

- Time of block A is not less than the Dagger era;
- Time of each input or output of block A is less than the time of block A;
- Each input or output of block A is a valid block;
- Sum of all input amounts of block B is less than power(2,64);
- Sum of all output amounts of block B plus its fee is less than power(2,64);

DAGGER
BY THE COMMUNITY

Algorithm : how to validate transaction

- If there is at least one input than sum of all inputs must be not less than sum of all outputs plus fee; otherwise sum of all outputs must be zero;
- For each input B of the block A there are public key K and input or output signature S in the block A and output signature T in the block B such that signature S is obtained from block A using key K and signature T is obtained from block B using the same key K (informal description: only owner of block B can withdraw money from it).
- Number of output signature fields must be even instead of number of input signature fields may be odd; in this case the last input signature field may be used as nonce which can be altered without rebuilding any signatures.

DAGGER
BY THE COMMUNITY

Algorithm : how to sort transactions

- Block referenced by a main block is ahead of block not referenced
- The smaller i-referenced block to the same common block is ahead
- The referenced block is ahead of linked block

DAGGER
BY THE COMMUNITY

Transport:

- Pool only broadcast blocks generated by itself
- Pool request other blocks

DAGGER
BY THE COMMUNITY

Security

- ECDSA secp256k1 for signing
- Semi-symmetric for transport

DAGGER
BY THE COMMUNITY

The Future

- PoW + DAG + Anti-asic + Anonymous Trading + Smart Contract

- Mobile Wallets
- Light Wallet
- Full Wallet

- **Golang version**
- **C++ Version**
- *Python Version*

DAGGER
BY THE COMMUNITY

How to Join & Help Community

Everyone related to XDAG is part of community

- Spread XDAG
- Discuss proposal
- Report issues
- Translation
- Contribute code

DAGGER
BY THE COMMUNITY

Thank you!

Thanks to all developers!

Evgeniy, sgaragagghu, trueserve,

Bill, Solar, Wendy, czslience, rubencm

Thanks to all Miners, Pool Owners and other contributors

Email: frozen@xdag.io

DAGGER
BY THE COMMUNITY

Dev QQ Group

WeChat Official Account